MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO

CC(T) DULCINEIA SANTOS SENNEJUNKER

T-SLOW: ALGORITMO PARA A DETECÇÃO DE ATAQUES SLOW DOS

Rio de Janeiro
2018

INSTITUTO MILITAR DE ENGENHARIA

CC(T) DULCINEIA SANTOS SENNEJUNKER

T-SLOW: ALGORITMO PARA A DETECÇÃO DE ATAQUES SLOW DOS

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientador: Prof. Anderson Fernandes Pereira dos Santos - D.Sc.

Rio de Janeiro
2018

c2018

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80 - Praia Vermelha
Rio de Janeiro - RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

006.3 Sennejunker, Dulcineia Santos
S478t T-SLOW: algoritmo para a detecção de ataques SLOW DOS / Dulcineia Santos Sennejunker, orientada por Anderson Fernandes Pereira dos Santos - Rio de Janeiro: Instituto Militar de Engenharia, 2018.

105p.: il.

Dissertação (Mestrado) - Instituto Militar de Engenharia, Rio de Janeiro, 2018.

1. Curso de Sistemas e Computação - teses e dissertações. 1. Lógica Fuzzy. 2. Algoritmo. I. Santos, Anderson Fernandes Pereira dos . II. Instituto Militar de Engenharia.

INSTITUTO MILITAR DE ENGENHARIA

CC(T) DULCINEIA SANTOS SENNEJUNKER

T-SLOW: ALGORITMO PARA A DETECÇÃO DE ATAQUES SLOW DOS

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientador: Prof. Anderson Fernandes Pereira dos Santos - D.Sc.

Aprovada em 21 de Maio de 2018 pela seguinte Banca Examinadora:

Prof. Anderson Fernandes Pereira dos Santos - D.Sc. do IME - Presidente

---

Prof. Cláudio Miceli de Farias - D.Sc. da UFRJ

Prof^{a}. Raquel Coelho Gomes Pinto - D.Sc. do IME

Rio de Janeiro
2018

À Marinha do Brasil, que possibilitou a realização desse trabalho.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus por me preparar e por ter colocado pessoas preciosas ao meu lado, que me ajudam em todos os momentos da minha vida.

Aos meus pais, José Carlos e Maria de Lourdes, pelo exemplo de carater, valores transmitidos e por terem me ensinado a trilhar pelos caminhos corretos, graças a Deus.

Ao meu orientador, Professor Doutor Anderson Fernandes Pereira dos Santos, pela paciência e dedicação com que me orientou, corrigindo-me quando necessário. Sua experiência e seus ensinamentos, com muita temperança, ajudaram-me a superar as dificuldades e desempenharam papel fundamental na realização deste trabalho.

À Marinha do Brasil que me possibilitou concretizar mais essa conquista pelo aprimoramento profissional. Assim, agradeço ao Capitão-de-Fragata (T) William Augusto Rodriguez de Souza, que incentivou-me a crescer profissionalmente, liberando-me para a realização do Curso de Mestrado e também, ao Capitão-de-Fragata (EN) Hélio Mendes Salmon pelos conselhos e confiança investida na realização desse trabalho.

Gostaria de agradecer ao Sr. Arash Habibi Lashkari pela disponibilização do CIC DoS Dataset, ao Sr. Hugo Gonzalez pelas respostas referentes aos ataques *Slow DoS* e também, à equipe do Grupo Alteryx Analytics, responsável pela disponibilização da licença para uso acadêmico da ferramenta Alteryx para as análises de *big data* realizadas nesse trabalho de pesquisa.

Meu agradecimento especial vai para o meu amado marido Kjeld, que assim como meus pais, sempre acreditou em mim. Com seu companheirismo, apoio e compreensão, esteve sempre por perto nos momentos mais difíceis e mais felizes também, graças a Deus.

“Nada é permanente, exceto a mudança. ”

HERÁCLITO DE ÉFESO

SUMÁRIO

LISTA DE ILUSTRAÇÕES 9

LISTA DE TABELAS 11

LISTA DE SIGLAS 12

1 INTRODUÇÃO 18
1.1 O Problema da Detecção *Slow DoS* 19
1.2 A Motivação e Hipóteses de Pesquisa 21

2 CONCEITOS BÁSICOS E ESTADO DA ARTE 24
2.1 Conceitos Básicos 24
2.1.1 Ataques Distribuidos de Negação de Serviço 24
2.1.2 Ataques de Negação de Serviço na Camada de Aplicação 26
2.2 Estado da Arte 29

3 MÉTRICAS PARA A DETECÇÃO *SLOW DOS* 34
3.1 A Métrica *Flags* 36
3.2 A Métrica *Omnia* 38
3.3 A Métrica *RTTR* 40

4 METODOLOGIA PARA O SISTEMA DE DETECÇÃO 43
4.1 Estimação do Tamanho das Janelas 45
4.1.1 Janelas de Tempo Estáticas 45
4.1.2 Janelas de Tempo Dinâmicas com a Lógica Fuzzy 46
4.2 O Previsor Média Móvel 59
4.2.1 Período Pré-Determinado para MMS 61
4.2.2 Período Variável para MMS por Erro Quadrático Médio 61
4.3 A Desigualdade Tchebycheff na Análise de Detecção 61
4.4 O Algoritmo *T-Slow* 63
4.4.1 A função *Flags* 63
4.4.2 A função *Omnia* 65
4.4.3 A função *RTTR* 66

5 ESTUDO DE CASO: *CIC DOS DATASET* 67
5.1 O *Dataset* 67
5.2 Resultados Obtidos 69
5.2.1 Resultados com a Métrica *Flags* 71
5.2.2 Resultados com a Métrica Omnia 72
5.2.3 Resultados com a Métrica RTTR 73
5.3 Análises de Desempenho 74
5.3.1 Janelas Estáticas com MM fixa 77
5.3.2 Janelas Estáticas com MM variável por EQM 78
5.3.3 Janelas Dinâmicas *fuzzy* com MM fixa 79
5.3.4 Janelas Dinâmicas *Fuzzy* com MM variável por EQM 81
5.4 Discussão da Metodologia 82
5.4.1 Análise Crítica 84

6 CONCLUSÃO 87

7 REFERÊNCIAS BIBLIOGRÁFICAS 92

8 APÊNDICES 99
8.1 APÊNDICE 1: Análise de Distribuição 100

9 ANEXOS 103
9.1 ANEXO 1: Prova da Desigualdade Tchebycheff 104

LISTA DE ILUSTRAÇÕES

FIG.1.1 Tipos de Ataques à Camada de Aplicação com base em Jazi et al. (2017). 20

FIG.2.1 Arquitetura *Agent-Handler* de um Ataque DDoS com base na taxonomia de Mirkovic e Reiher (2004). 25

FIG.2.2 Possibilidades de Ataque DDoS via camada OSI com base no CERT-U.S. (2014) 26

FIG.3.1 Distribuição da quantidade de IP por janelas de tempo. 35

FIG.3.2 Detalhes de uma requisição HTTP com base em Kurose e Ross (2014). 38

FIG.3.3 Detalhes de uma requisição HTTP com ausência dos marcadores CR LF CR LF para o último GET. Fonte: CIC-DoS (2017). 39

FIG.3.4 Tempo de ida e volta de pacotes com base em Kurose e Ross (2006) 41

FIG.4.1 Arquitetura para detecção de anomalias dos ataques *Slow DoS*. 44

FIG.4.2 Timeliness para detecção de anomalias. 45

FIG.4.3 Slowloris - Valores reais do discurso para a variável Max_RTT com base nos dados do CIC-DoS (2017) *Dataset*. 49

FIG.4.4 Slow Headers - Valores reais do discurso para a variável Max_RTT com base nos dados do CIC-DoS (2017) *Dataset*. 49

FIG.4.5 Slow Body - Valores reais do discurso para a variável Max_RTT com base nos dados do CIC-DoS (2017) *Dataset*. 50

FIG.4.6 Slow Read - Valores reais do discurso para a variável Max_RTT com base nos dados do CIC-DoS (2017) *Dataset*. 50

FIG.4.7 R.U.D.Y - Valores reais do discurso para a variável Max_RTT com base nos dados do CIC-DoS (2017) *Dataset*. 51

FIG.4.8 Slowloris - Valores reais do discurso para a variável Est_RTT com base nos dados do CIC-DoS (2017) *Dataset*. 52

FIG.4.9 Slow Body - Valores reais do discurso para a variável Est_RTT com base nos dados do CIC-DoS (2017) *Dataset*. 52

FIG.4.10 Slow Read - Valores reais do discurso para a variável Est_RTT com base nos dados do CIC-DoS (2017) *Dataset*. 53

FIG.4.11 R.U.D.Y - Valores reais do discurso para a variável Est_RTT com base nos dados do CIC-DoS (2017) *Dataset*. 53
FIG.4.12 Variável Max_RTT - Partição *Fuzzy*. 54
FIG.4.13 Variável Est_RTT - Partição *Fuzzy*. 54
FIG.4.14 Variável VJ - Partição *Fuzzy*. 56
FIG.4.15 Representação das Entradas Atuais com base em Ross (2009). 57
FIG.4.16 *Método da Média Ponderada* 58
FIG.4.17 Sistema *Fuzzy* pelo Método Mamdani com base em Zadeh (1976). 59
FIG.4.18 Gráfico comparativo para tomada de decisão. 60
FIG.4.19 Processamento da Métrica Omnia com a MMS com 30 observações. 61

FIG.5.1 Implementações e nomes atribuídos aos ataques do tipo *Slow Dos* presentes no CIC-DoS (2017) *Dataset*. 69
FIG.5.2 Distribuição Uniforme para o Tamanho das Janelas de Tempo. 70
FIG.5.3 Variação das Janelas de Tempo *Fuzzy*. 71
FIG.5.4 Gráfico para a distribuição dos valores máximos para *Flags* para janelas de tempo estáticas. 71
FIG.5.5 Distribuição dos somatórios de requisições incompletas (sem CRLF CRLF) por janela de tempo estática. 72
FIG.5.6 Gráfico para a distribuição dos valores da média RTTR por janela de tempo estática. 73

FIG.8.1 Gráfico de ajustamento à Distribuição Normal para a distribuição da média RTTR por janela de tempo estática. 101
FIG.8.2 Gráfico de ajustamento à Distribuição Normal para a distribuição de Flags_Prop por janela de tempo estática. 102

LISTA DE TABELAS

TAB.1.1 Países Alvos de Ataques à Camada de Aplicação com base nos dados da Akamai (2017). 18

TAB.2.1 Detecção de Ataques *Slow DoS* 32

TAB.3.1 Métricas F.O.R para a Detecção de Anomalias *Slow DoS* 42

TAB.4.1 Período de Captura de Tráfego no *Dataset* com base no CIC-DoS (2017). 46

TAB.4.2 Distribuição dos Intervalos para as Janelas de Tempo de 30 segundos com os dados do CIC-DoS (2017). 46

TAB.4.3 Tabela FAM das Regras de Produção *fuzzy* 57

TAB.5.1 Lista de Ataques com base no CIC-DoS (2017). 68

TAB.5.2 Valor de n que minimizará o erro de previsão - Estratégia 1. 74

TAB.5.3 Valor de n que minimizará o erro de previsão - Estratégia 2. 74

TAB.5.4 Avaliação do modelo preditivo para problema com duas classes com base em Faceli et al. (2015). 75

TAB.5.5 Janelas identificadas na Abordagem 1. 77

TAB.5.6 Matriz Confusão - Abordagem 1 77

TAB.5.7 Janelas identificadas na Abordagem 2. 78

TAB.5.8 Matriz Confusão - Abordagem 2 79

TAB.5.9 Janelas identificadas na Abordagem 3. 80

TAB.5.10 Matriz Confusão - Abordagem 3 80

TAB.5.11 Janelas identificadas na Abordagem 4. 81

TAB.5.12 Matriz Confusão - Abordagem 4 82

TAB.5.13 Quadro Comparativo das medidas de desempenho para as Abordagens 1 à 4 83

TAB.5.14 Quadro Comparativo de Desempenho entre Trabalhos sobre a Detecção de Ataques *Slow DoS* 84

LISTA DE SIGLAS

ACK	*Denial of Service*
ASV	*Adaptative Selection Verification*
CIC	*Canadian Institute for Cyber Security*
CPU	*Central Processing Unit*
CRLF	*Carriage Return Line Feed*
DDoS	*Distributed Denial of Service*
DDoSSim	*DDoS Simulator*
DNS	*Domain Name System*
DoS	*Denial of Service*
EQM	Erro Quadrático Médio
Est_RTT	RTT Estimado
FAM	*Fuzzy Associative Memory*
FIN ACK	Reconhecimento para uma Flag FIN
FN	Falso Negativo
FP	Falso Positivo
HTML	*HyperText Markup Language*
HTTP	*Hyper Text Transfer Protocol*
HULK	*HTTP Unbearable Load King*
ICMP	*Internet Control Message Protocol*
IP	*Internet Protocol*
KDD	*Knowledge Discovery in Databases*
KNN	*k-Nearest Neighbors algorithm*
Max_RTT	RTT máximo
MMS	Média Móvel Simples
NS	*Network Simulator*
OMNeT	Objective Modular Network Testbed
RFC	*Request for Comment*
RTT	*Round Trip Time*
RUDY	*aRe yoU Dead Yet?* (Tipo de ataque *Slow DoS*)
SDN	*Software-Defined Networking*
SYN	*Syncronize*
SYN ACK	Reconhecimento para uma Flag SYN
TCP	*Transmission Control Protocol*

T-Slow	*Tchebycheff-Slow*
UDP	*User Datagram Protocol*
URL	*Uniform Resource Locator*
VJ	Variação da Janela de Tempo
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

RESUMO

Recentemente, como consequência dos esforços de mitigação aos ataques de negação de serviço tradicionais, os ataques *Slow DoS* surgem como ameaça à garantia da disponibilização de serviços na *Web*.

Esses ataques são considerados preocupantes devido a furtividade do seu modo de operação, pois podem desabilitar a vítima lentamente e sem alarde. E, diferenciam-se dos ataques DDoS tradicionais, pois não causam mudanças abruptas no comportamento do tráfego de rede, como as que evidenciam um ataque volumétrico, tais como um aumento significativo de endereços IP em um dado instante no servidor alvo. A sofisticação dessa categoria de ataque motivou o presente trabalho, com a proposição de um algoritmo para a detecção de ataques *Slow DoS* à camada de aplicação, com foco na pronta detecção destes e com altas taxas de desempenho.

Durante a pesquisa, foram realizados estudos observacionais e experimentos para melhor avaliação do funcionamento das métricas propostas, e que compõem parte da estrutura do algoritmo principal chamado *T-Slow*. As métricas desenvolvidas nesse trabalho não destinam-se ao fornecimento de *thresholds* para a detecção, como por exemplo, a limitação de conexões que um servidor pode manipular ou uma limitação de várias conexões simultâneas que um cliente pode ter, mas sim à verificação do verdadeiro *status* dos atributos de rede em um dado momento.

Essas métricas procurarão detectar características ou padrões bem definidos nos traces maliciosos dos ataques lentos à camada de aplicação, mediante o aumento súbito no número de conexões, cabeçalhos das requisições incompletos ou leitura extremamente lenta das respostas da vítima. Dessa forma, as três funções do algoritmo *T-Slow*, cujos nomes referem-se às próprias métricas, chamadas *Flags, Omnia e RTTR* (F.O.R.), atuarão de forma paralela, porque cada uma das funções investiga uma característica do trace malicioso *Slow DoS*, que poderá evidenciar distorções nos valores históricos dos atributos do tráfego de rede.

Dois estudos observacionais foram realizados. O primeiro, para determinação do tamanho de janela adequado para a detecção e o segundo, para determinar o melhor tamanho de amostra para aplicação das métricas F.O.R. Com relação ao tamanho da janela, foi utilizada a lógica *fuzzy* para alteração dinâmica do intervalo de tempo analisado. Os resultados desse experimento foram observados e comparados aos resultados obtidos com as janelas de tempo estáticas. Com relação à escolha do melhor tamanho de amostra, para aplicação do previsor de anomalias aos resultados das métricas, foi utilizada a técnica do menor erro quadrático médio por ser uma técnica adequada às previsões de curto prazo. Assim, cada métrica poderá utilizar o tamanho de amostra mais adequado ao seu potencial. Os resultados desse experimento também foram observados e comparados aos resultados obtidos com as amostras, cujo tamanho não era variável, ou seja, comum à todas às métricas.

O principal diferencial deste trabalho, em relação a outros trabalhos sobre detecção de ataques *Slow DoS*, é a utilização da lógica *fuzzy* para a resolução do problema da estimação adequada das janelas de tempo, para a análise do tráfego de rede.

Outra observação importante é que, a literatura sobre algoritmos de aprendizado sugere que, estes sofrem com o problema de *overfitting* e o *underfitting* de dados durante o processo de treinamento. O *T-Slow* não sofre com tais problemas, porque o algoritmo faz uso do histórico de dados mais recentes da própria rede, com intervalos de dados definidos

pelo menor erro quadrático médio.

O núcleo do mecanismo de detecção utiliza a desigualdade Tchebycheff. Dado que não se conhece a distribuição de probabilidades das variáveis de tráfego envolvidas nesse trabalho, emprega-se esta desigualdade, pois ela se aplica a qualquer distribuição de dados. Com ela, foi possível alcançar baixa taxa de erro na classe positiva e, dentro do escopo do trabalho, alta acurácia em detecção. O algoritmo foi aplicado a um *dataset* de tráfego de rede real, que mescla tráfego de fundo ao tráfego de ataques à camada de aplicação, com os ataques *Slow Headers, Slowloris, Slow Body, R.U.D.Y. e Slow Read.*

ABSTRACT

Recently, as a result of mitigation efforts of traditional denial of service attacks, Slow DoS attacks appeared as a threat to the availability of web services. These attacks are considered to be worrisome due to their stealthily approach, disabling the victim slowly and without shew.

These attacks are different from traditional DDoS attacks, as they do not cause abrupt changes in the behavior of network traffic such as volumetric attacks, like the significant increase of active IP addresses in a given instant. The sophistication of this category of attacks motivated the present work, with the proposition of an algorithm for detecting Slow DoS attacks on the application layer, focused on prompt detection and high performance.

During the research, observational studies and experiments were carried out to evaluate the performance of the proposed metrics that comprise part of the main algorithm's structure, called T-Slow. The metrics developed in this work do not intend to provide thresholds for the detection, for example, a limit of connections that a server can handle or a limitation of multiple simultaneous connections that a client may have at the same time, but rather verify the true status of the attributes of the network at any given time.

The metrics will seek to detect characteristics and well-defined patterns of the malicious traces of the slow attacks on the application layer, like the sudden increase in the number of connections, headers of incomplete requisitions or extremely slow reading of the victim's responses. In this way, the algorithm T-Slow counts on three functions, whose names refer to the metrics themselves, called Flags, Omnia and RTTR (F.O.R.). Since each of the functions investigates a specific characteristic of traces of Slow DoS attacks that may show distortions in the network traffic attributes, they work in parallel.

Two important observational studies were carried out: The first one, to determine the appropriate analysis window size for the detection and the second, to determine the best sample size for application of the F.O.R. metrics. With respect to the determination of the appropriate analysis window size, Fuzzy Logic was applied to dynamically determine the time interval of analysis. The results of this experiment were compared to the results obtained with the static time windows. Regarding the choice of the best sample size, for application of the anomaly detection using the mentioned metrics, the method of the smallest average quadratic error was used, being suitable for short-term forecasts. This way, each metric uses the most suitable sample size. The results of this experiment were also compared to the results obtained with the samples whose size was not variable but common for all metrics.

The main difference of this work with respect to other works on the detection of Slow DoS attacks is the use of fuzzy logic to solve the problem of adequate estimation of the time windows for the analysis of network traffic.

Another important observation is that literature suggests that machine learning algorithms may suffer from data overfitting or underfitting during the training phase. The T-Slow algorithm does not face these issues since it uses the most recent data of the network, with data ranges defined by the smallest mean square error.

The core of the detection mechanism uses the Tchebycheff inequality. The distribution of the traffic variables used in this work is not known, however the inequality of Tchebycheff can be applied to any distribution. This way it was possible to achieve low rates of error in the positive class and high rates if detection accuracy, within the scope

of the work. The algorithm was applied to a dataset of real network traffic that contains background traffic merged with attack traffic to the application layer, in specific the attacks Slow Headers, Slowloris, Slow Body, R.U.D.Y. e Slow Read.

1 INTRODUÇÃO

A negação de serviço, ou DoS (*Denial of Service*), é uma técnica pela qual um atacante utiliza um computador para tirar de operação um serviço, um computador ou uma rede conectada à Internet (CGI.BR, 2012a). Quando utilizada de forma coordenada e distribuída, ou seja, quando um conjunto de computadores é utilizado no ataque, recebe o nome de ataque distribuído de negação de serviço, ou DDoS (*Distributed Denial of Service*) (CGI.BR, 2012a).

Os ataques de negação de serviço são preocupantes, porque impedem que usuários legítimos acessem os serviços disponibilizados em provedores. Em 2009, ataques cibernéticos ao Irã mostraram que esse tipo de ataque é uma séria ameaça não apenas para serviços de internet, mas também para governos (CAMBIASO et al., 2012).

Mecanismos para mitigar os efeitos dos ataques DDoS, tais como os baseados em ASV *(Adaptative Selection Verification)* (KHANNA et al., 2012) para ataques distribuídos de negação de serviço (DDoS) demonstraram resultados positivos. Recentemente, em decorrência dos esforços da comunidade científica para a mitigação dos ataques de negação de serviço, os atacantes que, anteriormente, visavam à camada de rede ou transporte, voltaram sua atenção à camada de aplicação.

De acordo com a Tabela 1.1, observa-se que, no quarto trimestre de 2017, o Brasil ocupou o segundo lugar no *ranking* dos países destinos neste tipo de ataque.

TAB. 1.1: Países Alvos de Ataques à Camada de Aplicação com base nos dados da Akamai (2017).

Posição	País	Nº de Ataques
1	EUA	238.643.360
2	Brasil	21.900.411
3	Reino Unido	19.385.710
4	Canadá	17.459.934
5	Alemanha	13.432.389

Observa-se que os ataques de negação de serviço à camada de aplicação tornaram-se populares (JIANG et al., 2017) e inserem-se no cenário da guerra cibernética tal como os ataques DDoS volumétricos ou por inundação (MIRKOVIC; REIHER, 2004) à camada de transporte. Além dos ataques tradicionais ou volumétricos à camada de aplicação

existem também os ataques lentos e de baixa taxa com relação ao número de conexões. Os ataques de baixa taxa caracterizam-se por transmitir à vítima, estrategicamente, pequenas quantidades de tráfego, e os ataques lentos caracterizam-se por explorar os parâmetros de tempo, no lado do servidor, ao enviar e receber tráfego mais lento que o normal (JAZI et al., 2017).

A principal técnica utilizada por estes ataques objetiva atingir seu alvo mediante vulnerabilidades inerentes ao protocolo HTTP (*Hyper Text Transfer Protocol*), como por exemplo, o tempo de resposta para processamento de uma requisição.

1.1 O PROBLEMA DA DETECÇÃO *SLOW DOS*

Define-se o ataque como lento, porque o atacante atinge seu objetivo, a negação de serviço, não como em um ataque DDoS tradicional, mas sim pelo envio de pacotes maliciosos ao alvo, que dissimuladamente e lentamente, consumirão recursos importantes do alvo, destinados aos usuários legítimos. Tais ataques beneficiam-se da conexão persistente do protocolo HTTP para manter pelo maior tempo possível a conexão com o servidor, enquanto esgotam-se seus recursos de memória e CPU. Assim, o ataque assume sua característica furtiva, porque antes que o tempo de espera expire, outra requisição é enviada ao servidor pela mesma conexão com o objetivo de reiniciar o relógio de tempo de espera no lado do servidor.

Segundo Bhuyan et al. (2014), obter mecanismos de detecção mais rápidos é o principal problema enfrentado na detecção de ataques DDoS, porque esquemas de detecção consomem recursos da vítima e isso afeta a capacidade de acurácia. Outro problema de difícil solução, com relação aos ataques lentos e de baixo volume de tráfego, é detectá-los com alta acurácia e baixas quantidades de falsos positivos. Os tipos de ataques DoS na camada de aplicação são ilustrados pela Figura 1.1.

O desenvolvimento de um mecanismo de defesa é uma tarefa difícil, porque demandam conhecimentos pertencentes à diversas áreas de pesquisa. Assim sendo, os segmentos de pesquisa sobre o tema DDoS estão divididos, dentre outros, sob os enfoques da identificação, detecção ou mitigação do ataque.

Uma fase importante e decisiva na pesquisa de qualquer sistema de detecção é a avaliação da *performance* (desempenho) de detecção que buscará evidenciar detalhadamente os resultados alcançados.

Para alcançar este objetivo, métodos estatísticos, *soft computing* (redes neurais, lógica *fuzzy*), sistemas baseados em conhecimento (KDD), ou métodos baseados em aprendizado

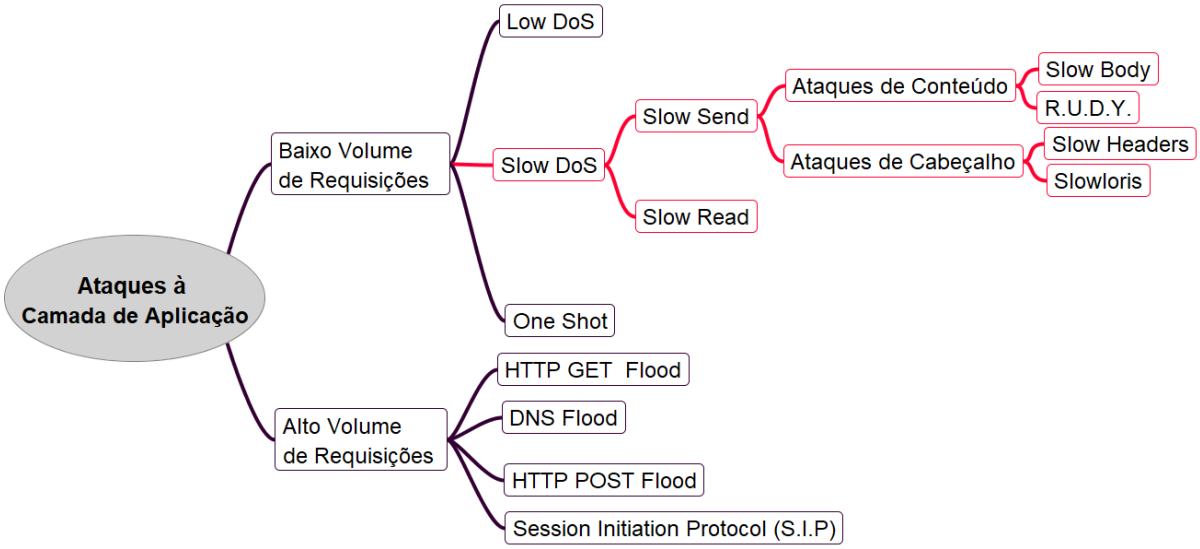


FIG. 1.1: Tipos de Ataques à Camada de Aplicação com base em Jazi et al. (2017).

de máquina e *data mining* são utilizados para processamento dos dados do tráfego de rede.

Um mecanismo de defesa contra ataques DoS não pode simplesmente ser julgado com base na sua performance com relação a um *dataset* fixo contendo um tráfego normal e poucos pacotes de ataque, porque espera-se que os sistemas de detecção de ataques distribuidos de negação de serviço em tempo real sejam escaláveis, ou seja, que possam atuar também em redes reais de alta velocidade (BHUYAN et al., 2014).

Assim, sob o enfoque da detecção de ataques, além do parâmetro ferramenta, um outro parâmetro que deverá ser considerado no estudo da detecção do ataque DDoS é o tipo do *dataset*. Segundo Bhuyan et al. (2014), os *datasets* podem ser classificados em:

a) *Benchmark*: *datasets* preparados por uma rede empresarial com os rótulos normal ou com ataque. Nesta categoria, existem poucos *datasets* disponíveis;

b) *Dataset* Simulado: uma outra alternativa são os simuladores de ambiente de rede, tais como NS-3 (RILEY; HENDERSON, 2010), Qualnet (SIMULATOR, 2011), OMNeT++ (VARGA; HORNIG, 2008). Mas, como simuladores que são, é muito difícil certificar que o tráfego gerado está bem correlacionado a uma rede de tráfego real; e

c) *Dataset* Privado: é a melhor abordagem para testar qualquer sistema de detecção ou detecção de ataque DDoS com um grande número de *hosts* e componentes de rede. Assim, um *dataset* poderia ser gerado com o tráfego normal e outro com o tráfego de ataques, capturando ambos com ferramentas para redes de alta velocidade.

Para uma boa escolha do *dataset*, deve-se observar se este possui um tráfego de rede real corretamente capturado. Esse fator é fundamental na avaliação de um novo esquema de detecção, para que tenha bom desempenho em uma rede real. No entanto, ainda existe grande dificuldade na obtenção de *dataset* privado para a pesquisa de detecção de ataques.

Assim, os seguintes problemas de pesquisa foram abordados:

Questão de Pesquisa 1. *Quais métricas são competentes na detecção das anomalias inerentes ao ataques lentos à camada de aplicação (Slow DoS)?*

Questão de Pesquisa 2. *O algoritmo proposto obteve boas taxas de desempenho?*

1.2 A MOTIVAÇÃO E HIPÓTESES DE PESQUISA

A motivação para o estudo do tema apresentado tem origem na complexidade da investigação, em decorrência da sofisticação com que os ataques lentos de negação de serviço conseguem desabilitar servidores. A detecção das anomalias proporcionadas por esses ataques é uma tarefa desafiadora devido à semelhança do tráfego de ataque com o tráfego normal. O tráfego de ataque pode passar despercebido pelo radar dos detectores de anomalia. A sofisticação dos ataques contra os esforços de mitigação serve como motivação para engajamento nessa batalha da guerra cibernética.

Normalmente, na pesquisa sobre a detecção dos ataques de negação de serviço, costuma-se a dividir o tráfego em janelas de tempo com intervalos regulares ou estáticos, visando a detecção das anomalias decorrentes desses ataques. Neste caso, o problema é que alguma alteração nos atributos do tráfego poderá passar despercebida pelo mecanismo de detecção.

Outra questão envolvendo a detecção de anomalias é a escolha do período de tráfego a ser analisado, ou seja, de que forma deveria-se selecionar o intervalo de dados, para que este seja representativo do tráfego normal, a fim de permitir a realização de inferências com relação ao tráfego de ataque.

Para a resolução desses problemas, investigou-se se o algoritmo proposto seria capaz de contribuir para a seleção de *trace* malicioso, comprovando as seguintes hipóteses:

Hipótese 1. *A lógica fuzzy, aplicada na estimação do tamanho das janelas, contribui para o processo de detecção?*

Hipótese 2. *A estimação do histórico da rede realizado com a técnica das médias*

móveis com intervalos determinados pelo Erro Quadrático Médio é eficiente?

Hipótese 3. *O intervalo utilizado pela desigualdade Tchebycheff é apropriado para a estimação da distribuição do tráfego de ataque?*

Geralmente, a escolha por janelas estáticas ocorre mediante empirismo, onde escolhe-se o período de tempo que oferecer melhor taxa de detecção (JAZI et al., 2017). A detecção dos ataques de negação de serviço baseada em janelas estáticas podem ocultar diferenças ou mudanças de comportamento dos atributos do tráfego de rede, mascarando o *trace* malicioso dos ataques *Slow DoS*.

Esses ataques são difíceis de serem detectados, justamente por assemelhar-se ao tráfego normal. Portanto, dispor das métricas adequadas, para a detecção desses ataques, é uma condição fundamental para o sucesso do mecanismo de detecção. Visando a resolução desses problemas, planejou-se um algoritmo que pudesse realizar a detecção das anomalias referentes aos ataques *Slow DoS*.

Essa dissertação tem por objetivo principal, o desenvolvimento de um algoritmo, chamado *T-Slow*, composto por um comitê de métricas de votação uniforme, onde a aplicação da desigualdade Tchebycheff indicará a suspeita de *trace* malicioso do tipo *Slow DoS* com resultados assertivos.

A expectativa é que, com os resultados obtidos com a aplicação do *T-Slow*, o presente trabalho contribua para os estudos na área de detecção dos ataques lentos e, sobretudo, fomentar a discussão sobre a melhoria nas taxas de desempenho utilizando o previsor média móvel e quais destas têm maior influência na detecção.

Destacam-se as seguintes contribuições esperadas:

- A divulgação das recentes descobertas sobre a detecção das anomalias de tráfego com base nas características de ataques lentos de negação de serviço, apresentando uma discussão sobre as variáveis de tráfego que sejam úteis e heurísticas que contribuam para a detecção do *Slow DoS* e;

- Analisar os resultados de previsão com o algoritmo *T-Slow*, mediante quatro abordagens distintas de tratamento dos dados com o *dataset* utilizado. Essas abordagens diferem quanto a avaliação das janelas de tempo e quanto à amostragem dos dados para análise a curto prazo, com vistas à adequada detecção de *trace* malicioso para alertar, tempestivamente, sobre a ocorrência de anomalias do tipo *Slow DoS*.

Na seção 2 desse trabalho, descreve-se alguns conceitos básicos e expõe-se o estado da arte. Essa contextualização é necessária para o entendimento da seção 3, que tratará dos

atributos do tráfego de rede importantes para a detecção dos ataques e compreensão das métricas a serem utilizadas pela metodologia de detecção, que será proposta na seção 4. Na seção 4, é proposta a metodologia e o algoritmo principal *T-Slow* com as respectivas funções. Na seção 5, é apresentado um estudo de caso, cujos resultados das abordagens para a detecção dos ataques *Slow DoS* são analisados, segundo medidas de desempenho, proporcionando a discussão da metodologia. Na seção 6, é apresentada a conclusão e indicação de trabalhos futuros.

2 CONCEITOS BÁSICOS E ESTADO DA ARTE

Para um melhor entendimento sobre as etapas da metodologia proposta é necessário contextualizar o assunto detecção DDoS. Primeiramente, serão apresentados alguns conceitos básicos, e na seção posterior, um breve histórico sobre as abordagens de detecção.

2.1 CONCEITOS BÁSICOS

A taxonomia dos ataques expõe os critérios de classificação dos ataques e os meios de defesa empregados (MIRKOVIC; REIHER, 2004; BHUYAN et al., 2014). Algumas taxonomias sobre detecção dos ataques de negação de serviço classificam os ataques em função da quantidade de máquinas empregadas na ação, ou seja, um ataque é classificado como DDoS quando várias máquinas são empregadas e DoS quando apenas uma máquina é empregada (GUPTA; BADVE, 2017).

2.1.1 ATAQUES DISTRIBUIDOS DE NEGAÇÃO DE SERVIÇO

A negação de serviço, ou DoS, é uma técnica pela qual um atacante utiliza um computador para tirar de operação um serviço, um computador ou uma rede conectada à Internet (CGI.BR, 2012b).

Conforme mencionado no capítulo anterior, o ataque distribuído de negação de serviço (DDoS) ocorre de forma coordenada e distribuída, quando um conjunto de computadores é utilizado no ataque(CGI.BR, 2012b).

A motivação dos atacantes varia muito em função da experiência, ideologia ou intenções. Independentemente do motivo, eles exploram vulnerabilidades nos sistemas ou equipamentos das vítimas para então, executar o ataque. Em função de como ocorre a negação de um serviço aos usuários legítimos, esta é enquadrada em duas categorias: ataques DDoS por inundação ao nível da camada de rede ou transporte e ataques DDoS por inundação ao nível da aplicação.

Na primeira categoria, observa-se o envio brutal de mensagens, aparentemente legítimas, a um sistema alvo, consumindo recursos importantes da vítima, tais como memória, CPU e largura de banda do equipamento alvo. Para atingir o objetivo, uma das formas utilizadas pelos ataques de inundação é a exploração de protocolos, onde os atacantes exploram recursos específicos ou vulnerabilidades de alguns protocolos da vítima para

consumir quantidades excessivas de seus recursos. Mensagens complexas ou grandes podem exigir um processamento longo que absorve ciclos de CPU e ocupam largura de banda (MIRKOVIC; REIHER, 2004). Assim, o objetivo do ataque é atingido através do envio de grandes volumes de pacotes que ocupam uma proporção significativa de largura de banda disponível, consumindo este recurso considerado crítico num serviço de rede (PENG et al., 2007). Na segunda categoria, os ataques DDoS ao nível da aplicação, geralmente consomem menos largura de banda, no entanto, são mais furtivos no seu modo de operação em comparação com os ataques volumétricos ao nível da camada de transporte, pois os ataques à camada de aplicação assemelham-se ao tráfego normal (ZARGAR et al., 2013).

O ataque DDoS é uma extensão comum dos ataques de negação de serviço em que o poder de ataque aumenta com inúmeras fontes de ataque (máquinas "*zumbis*" em uma *botnet*) que estão sob o controle de um atacante. A rede de ataque DDoS pode seguir uma arquitetura do tipo *Agent – Handler* que é composta pelo atacante (cliente), mestres (*handlers*) e agentes (*agents*) (BHUYAN et al., 2014), conforme Figura 2.1. O atacante comunica-se com todo o sistema de ataque DDoS e também é chamado de *botmaster*.

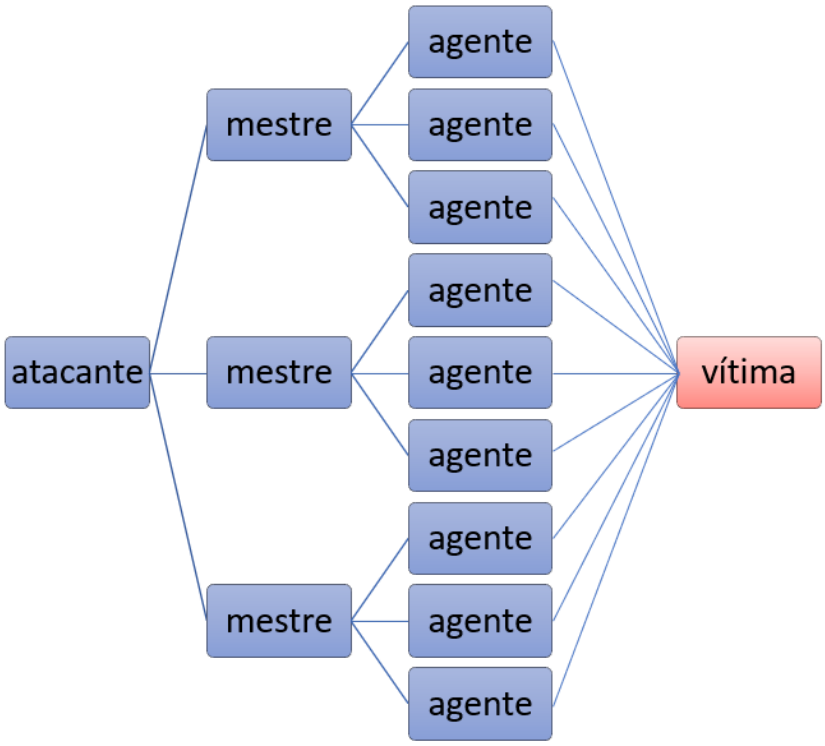


FIG. 2.1: Arquitetura *Agent-Handler* de um Ataque DDoS com base na taxonomia de Mirkovic e Reiher (2004).

Os mestres são aplicações maliciosas escondidas num equipamento, através do qual o *botmaster*, atacante executa o ataque. Os mestres são capazes de controlar inúmeros

agentes (*HOST* comprometidos). Já os agentes estão em sistemas comprometidos que eventualmente executarão um ataque C&C (comando e controle). Os usuários e proprietários dos sistemas comprometidos, geralmente não tem ciência da situação. Os agentes são capazes de gerar cadeias de pacotes em direção a vítima.

Dois fatos são considerados de grande importância. O primeiro é que as máquinas "agentes", geralmente são externas à rede própria da vítima e externas à rede do invasor. O segundo é que, consequentemente, é difícil para a vítima responder eficientemente contra o ataque e obter o *trace* do invasor (BHUYAN et al., 2014).

Os agentes podem ser cooptados por intermédio de brechas na segurança de um sistema operacional, um componente comum de *middleware* ou protocolo de rede, bem como programas de aplicativos. Uma classificação mais ampla de ataques de negação de serviço quanto a atuação na camada OSI é apresentada pela Figura 2.2

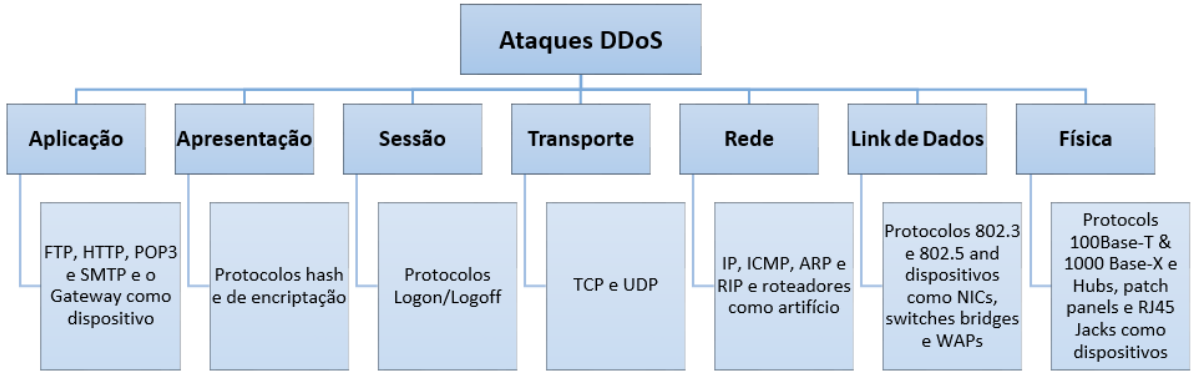


FIG. 2.2: Possibilidades de Ataque DDoS via camada OSI com base no CERT-U.S. (2014)

Dentre as diversas modalidades de negação de serviço, observa-se que a maioria explora as características dos protocolos. Dentre os ataques conhecidos que atingem a camada de transporte, cita-se os ataques *SYN Flood*, o *Smurf Attack*, as inundações UDP, sendo que os dois últimos ataques geram, respectivamente, muitos pacotes ICMP e UDP a fim de esgotar a capacidade de conexão do *link* de rede da vítima e nos ataque *SYN Flood*, os invasores enviam tantas solicitações de conexão para um servidor que os usuários não podem se conectar a esse servidor (OHSITA et al., 2006). E, são exemplos de ataques à camada de aplicação os seguintes: ataques de inundação HTTP POST ou GET, Slowloris, R.U.D.Y, Slow Read e Fragmentação HTTP (ZARGAR et al., 2013).

2.1.2 ATAQUES DE NEGAÇÃO DE SERVIÇO NA CAMADA DE APLICAÇÃO

O protocolo HTTP (*Hypertext Transfer Protocol*) utiliza o protocolo da camada de transporte TCP para a transmissão confiável de dados. Dentre as diversas modalidades de

negação de serviço, observa-se que a maioria explora as características dos protocolos no nível da camada de transporte tais como UDP e TCP/IP (OHSITA et al., 2006). Mas, com os recentes esforços à mitigação dos ataques DDoS tradicionais, os atacantes mais uma vez se concentram na realização de ataques à camada de aplicação.

As duas vertentes mais conhecidas desse tipo de ataque são os de alto volume de requisições, semelhantes aos ataques DDoS tradicional (*High Volume* ou *Flood*) e o baixo volume de requisições (*Low Volume*) com tráfego malicioso enviado à vítima em pequenas porções.

Nos ataques com alto volume de requisições, denominados também de ataques de força bruta, o atacante instrui *hosts* comprometidos a enviarem muitas requisições HTTP (tantas quanto forem possível) à vítima para consumir seus recursos tais como CPU e memória (JIANG et al., 2017). Como exemplo desses ataques cita-se o HTTP *Flood* e o HULK (*HTTP Unbearable Load King*).

Os ataques de baixa taxa classificam-se nas seguintes modalidades: Ataques *Low Rate*, *Slow Rate* e *One Shot* (JAZI et al., 2017).

Embora tenham sido introduzidos novos recursos ao protocolo HTTP 2.0, tais como mecanismo de controle de fluxo e algoritmo de compactação de cabeçalho, para mitigação dos transtornos do HTTP 1.1, segundo (JIANG et al., 2017), tais recursos também podem ser explorados por atacantes para empreender um ataque DDoS.

O protocolo HTTP 1.1 da camada de aplicação utiliza o TCP como protocolo da camada de transporte para transmissões confiáveis. No entanto, segundo Tripathi e Hubballi (2018) o protocolo HTTP 2.0, apresentou mais deficiências do que seu predecessor no que tange aos ataques *Slow DoS*.

O escopo dessa dissertação relaciona-se somente ao protocolo HTTP 1.1 e aos ataques DDoS com baixo volume de requisições *Slow DoS*.

Face a diversidade dos tipos de técnicas de ataques de negação de serviço, surgiram diferentes nomenclaturas para os ataques *Slow DoS*. Nessa subseção, pretende-se alinhar os vários termos encontrados na literatura até o momento. Para melhor entendimento dos processos de análise de tráfego de rede, serão apresentadas algumas definições importantes.

Segundo Gonzalez et al. (2014), as abordagens das linhas de pesquisa de ataques à camada de aplicação podem seguir os seguintes rumos de investigação: o desvio de comportamento normal de *web pages* de usuários, o monitoramento de requisições de clientes baseadas em *puzzle*, onde propõe-se um *puzzle* para solucionar e detectar um potencial ataque DoS ou podem basear-se nas características do tráfego de rede. Nesta última abordagem, destacam-se: os estudos para diferenciar o ataque DoS, do evento

Flash Crowd, mediante monitoramento do IP; a pesquisa relacionada aos ataques do tipo *Low Rate*, cujo foco reside nas características do tráfego de rede (pacotes de chegada); e, ataques do tipo *Slow Rate*, também conhecidos como *Slow DoS*, com foco nas propriedades comuns aos protocolos da camada de aplicação. Nestas abordagens, o objetivo é revelar ou prever padrões específicos do ataque. O presente trabalho enquadra-se na linha de pesquisa sobre as características do tráfego de rede para detecção de anomalias referentes aos ataques *Slow DoS*.

A maneira pela qual o cliente malicioso atua, ocorre basicamente em três variações: ataques de cabeçalho (*Slowloris* ou *Slow Headers*), ataques de conteúdo (*Slow Body* ou *R.U.D.Y.*), sendo estes também conhecidos como ataques *Slow Send* e o ataque *Slow Read* (JAZI et al., 2017).

Segundo (JAZI et al., 2017), o *Slow Send* é um ataque com foco no cabeçalho ou corpo da requisição (às vezes chamadas como ataques lentos de cabeçalhos ou ataques lentos de conteúdo (*body*)). O ataque lento de cabeçalho pode ser executado com uma variação popular implementada pela ferramenta *Slowloris* (RSNAKE; LEE, 2009) que oferece solicitações HTTP parciais (apenas com métodos GET ou POST) enviadas em intervalos regulares para manter a conexão; e com uma implementação mais geral que permite o uso de vários métodos que podem ser encontrados em *Slowhttptest* (SHEKYAN, 2013). O ataque lento de conteúdo começa com um cabeçalho de mensagem HTTP legítimo e, em seguida, continua enviando uma carga HTTP GET ou POST a um ritmo lento (por exemplo, 1byte/1 min) (CHEE; BRENNAN, 2010) (NAJAFABADI et al., 2016). O ataque lento de conteúdo *(body)* pode ser executado com a implementação *R.U.D.Y.* (NAJAFABADI et al., 2016) (RAZ, 2010) que requer presença de formulários em uma página da *Web* ou com *Slowhttptest* que implementa versão genérica sem a necessidade de formulários da Web (JAZI et al., 2017).

Com o *Slow Read*, o atacante envia as mensagens de requisição legítimas e lê as respostas do servidor lentamente. Segundo (PARK et al., 2014), o ataque explora o controle de fluxo do TCP, ou seja, o atacante envia um pedido legítimo após o *three way handshake* e, depois o atacante anuncia o tamanho da janela menor que o de costume, fazendo reduzir a operação de resposta HTTP. Assim, o servidor envia dados lentamente para o cliente, mantendo seus *sockets* abertos. O servidor continua investigando o cliente a fim de verificar seu tamanho de janela de recepção, enquanto o cliente sempre o adverte sobre o pequeno tamanho de janela, diminuindo assim a taxa de transferência. Quanto maior o tamanho do arquivo, mais tempo levará para concluir essas conexões.

O ataque *One Shot* causa danos à vítima com uma única requisição ou conexão

para consumir as capacidades de recursos da vítima. É um *exploit* para vulnerabilidade específica de um protocolo da camada de aplicação, como por exemplo o ataque *Apache Range Header* (JAZI et al., 2017).

Esse trabalho não trata o ataque na modalidade *Slow DoS* como um ataque DDoS tradicional, visto que tais ataques beneficiam-se de vulnerabilidades na segurança, comandando várias máquinas infectadas normalmente mal-intencionadas (*bots*) (SILVA et al., 2013), a enviarem uma grande quantidade de tráfego de ataque. Ao contrário, refere-se ao *Slow DoS* como um ataque DDoS não tradicional, que faz juz ao termo DDoS, fundamentalmente, porque realiza a abertura de inúmeras conexões para concretizar a negação do serviço de maneira furtiva. Essa característica mostra que este é um ataque que, diferentemente dos ataques DDoS tradicionais, atua sob o radar de detecção, sendo muito difícil de ser detectado.

2.2 ESTADO DA ARTE

Com a evolução das novas e sofisticadas ferramentas de ataque DDoS, novos mecanismos de defesa têm sido propostos. Para que a arquitetura de defesa seja eficiente, o mecanismo de defesa deverá atuar em função do local onde o atacante põe em prática o seu ataque (BHUYAN et al., 2014), ou seja, onde o ataque ocorrer, o mecanismo de defesa deverá atuar.

A abordagem de detecção *Victim-End* (MIRKOVIC; REIHER, 2004) ou *Victim Network* (BHUYAN et al., 2014), geralmente é empregada nos roteadores de rede da vítima. Os ataques DDoS podem ser detectados nos roteadores da vítima (BHUYAN et al., 2014), quando se trata de ataques volumétricos, por conta da elevada taxa de consumo dos recursos.

Segundo Osanaiye et al. (2016) os métodos de detecção de anomalia classificam-se em: *machine learning, Data Mining,* Inteligência Artificial, métodos estatísticos e baseados em classificadores. A principal vantagem desses métodos é o aprendizado do comportamento esperado mediante observações sem conhecimento prévio das atividades normais do sistema alvo (GUPTA; BADVE, 2017), como por exemplo demanda de serviço entre outras.

Os mecanismos de detecção de ataques *Slow DoS* mais conhecidos são aqueles que atuam na máquina vítima, mediante configurações prévias de administradores de redes ou *softwares* de gerenciamento de rede que utilizam o conhecimento da própria rede para a defesa.

Na literatura, os trabalhos sobre SDN (*Software-Defined Networking*) mostram-se promissores mediante um conceito de gerenciamento de rede que pode oferecer defesa eficaz contra ataques DDoS (HONG et al., 2017). Segundo Yan et al. (2016), é possível utilizar o mecanismo de defesa SDN contra ataques volumétricos à camada de aplicação, porque ataques de inundação, como o HTTP *Post*, têm o mesmo impacto nos servidores. Este autor ressalta o uso da arquitetura SDN somente para os ataques por inundação.

De acordo com Zargar et al. (2013), os mecanismos de defesa contra os ataques de inundação DDoS à camada de aplicação, com relação ao local onde atuam, pode ser enquadrados em duas categorias:

- *Destination-based*: quando estão no lado do servidor, como por exemplo DDoS-*Shield*, que utiliza métodos estatísticos na detecção; e
- Híbridos, quando empregam colaboração entre clientes e servidores, como por exemplo, quando empregam o *Completely Automated Public Turing test to tell Computers and Humans Apart* (CAPTCHA).

O presente trabalho destaca-se dos demais trabalhos na área de detecção de ataques *Slow DoS*, porque propõe um método matemático para a determinação adequada da variação do tamanho dos intervalos de tempo para análise de tráfego, ou seja, a variação no tamanho das janelas de tempo. Portanto, mediante a utilização da lógica *fuzzy*, o presente trabalho não baseia-se em empirismo. Diferentemente, Tripathi e Hubballi (2018) utilizaram janelas de tempo de dez minutos para a detecção de ataques *Slow Rate* em um *testbed*, cujo o servidor era o Apache 2.4.23 para requisições HTTP/2 sem nenhum motivo pré-determinado para tal. Jazi et al. (2017) propôs um algoritmo de somas cumulativas não paramétricas para a detecção de ataques *Slow DoS* em janelas de tempo de quinze, vinte e trinta segundos, comparando esses resultados segundo diferentes técnicas de amostragem de tráfego de rede para avaliação do impacto da amostragem nas taxas de detecção. A janela de tempo de trinta segundos foi a que apresentou melhores resultados e por isso foi a escolhida.

Alguns trabalhos não tratam dos ataques *Slow DoS* com maior amplitude, tratando apenas determinados tipos desse ataque (HONG et al., 2017; NAJAFABADI et al., 2016; KATKAR et al., 2015). Nesse sentido, o presente trabalho destina-se à detecção das anomalias referentes a todos os tipos de ataques *Slow DoS*: ataques de cabeçalho, ataques de conteúdo e ataques *Slow Read*. Em Hong et al. (2017), é proposto o SDN-*Assisted Slow HTTP DDoS Attack Defense Method* para detectar apenas os ataques *Slowloris* e *Slow* HTTP POST. Neste método, o servidor solicita ao controlador SDN, a verificação

do tráfego suspeito. Como por exemplo, quando um servidor *Web* recebe uma requisição HTTP incompleta em uma situação onde o número de conexões abertas no servidor *Web* excede o limiar pré-determinado de conexões concorrentes sendo processadas, o tráfego é considerado tráfego de ataque (HONG et al., 2017). Os autores utilizaram o simulador NS-3. Outra abordagem que avaliou a detecção para apenas um tipo de ataque *Slow DoS* foi proposta por Najafabadi et al. (2016), que utilizou aprendizado de máquina para a determinação das *features* ou atributos mais importantes para a detecção do ataque de conteúdo denominado R.U.D.Y. (*"aRe yoU Dead Yet?"*), sendo este o nome dado à implementação do ataque. Foi utilizado um comitê de 10 métodos para selecionar as *features* mais importantes para a classe rotulada, dentre eles, a estatística Kolmogorov-Smirnov, Curva ROC, Qui-Quadrado, ganho de informação entre outras. As *features* selecionadas (sete *features*) foram submetidas a três classificadores (KNN, C4.5N e c4.5D) e um outro conjunto de dados com o tamanho de *features* original também foi submetido ao mesmo grupo de classificadores. Não houve diferença significativa na aplicação dos métodos para redução de dimensionalidade. O *dataset* utilizado foi o produzido por Wheelus et al. (2014) denominado SANTA *dataset*. Este trabalho demonstrou a preocupação na seleção das *features* mais importantes a serem utilizadas por um classificador. Hirakawa et al. (2016) propôs limiares para o número de conexões por IP com base no número de conexões. E, Katkar et al. (2015) desenvolveu um sistema detector de intrusão de ataques DDoS contra servidores utilizando um classificador Bayesiano. Diferentemente do propósito do SDN, que monitora o tráfego, Dantas et al. (2014), desenvolveu um sistema de defesa baseado em verificação seletiva (*Adaptative Selective Verification*), denominado SeVen para mitigação de ataques DDoS à camada de aplicação, o qual se mostrou eficiente para a mitigação dos ataques HTTP POST e *Slowloris*.

Apresenta-se um quadro comparativo com o resumo dos mais recentes trabalhos com foco na detecção de ataques *Slow DoS* na Tabela 2.1.

Recentemente surgiram mais trabalhos com foco na nuvem (YAN et al., 2016; GUPTA; BADVE, 2017). Segundo Shameli-Sendi et al. (2015), a tecnologia SDN trouxe novas perspectivas para a mitigação de ataques DDoS na nuvem.

Também, surgiram novos *datasets* (JAZI et al., 2017; SHARAFALDIN et al., 2018; SHIRAVI et al., 2012) com tráfego de ataque *Slow DoS*, comprovando a importância da pesquisa aos ataques à camada de aplicação na modalidade de baixa taxa de requisições. Mas, a criação de *datasets* é uma tarefa bastante difícil e ainda existe pouca disponibilidade.

Outra vertente de pesquisa, são as novas variantes de ataques *Slow DoS*. Cambiaso

TAB. 2.1: Detecção de Ataques *Slow DoS*

Quadro Comparativo sobre Detecção de Ataques *Slow DoS*				
Autor	Método Empregado	Local do Mecanismo	*Dataset*	Ataque Alvo
(JAZI et al., 2017)	Algoritmo baseado em somas cumulativas	Servidor	CIC DoS	*Slow Headers, Slowloris, Slow Body, R.U.D.Y., Slow Read, H.U.L.K, DDoSSim e Goldeneye*
(HONG et al., 2017)	*SDN-Assisted*	Rede	NS-3	*Slowloris e Slow Post*
(HIRAKAWA et al., 2016)	Limitação do número de IP's	Servidor	Privado	*Slow Headers, Slow Post e Slow Read*
(NAJAFABADI et al., 2016)	*Machine Learning* (KNN e Árvore de Decisão)	Rede	SANTA *Dataset*	R.U.D.Y.
(KATKAR et al., 2015)	*Machine Learning* (Classificador Naive Bayes)	Servidor	Privado	*Syn Flood, Slowloris e Slow Read*

et al. (2017) desenvolveu e avaliou um novo tipo de ataque *Slow DoS*, o *Slowcomm*. O autor preocupou-se na exploração do tema "Internet das Coisas", dado que alguns dispositivos possuem poucos recursos de segurança. O novo ataque foi comparado ao *Slowloris*, sendo mais dificil de detectar que este. Foram propostas variações que podem atuar principalmente em ambientes móveis (CAMBIASO et al., 2016) e na nuvem (AMEYED et al., 2015), visto que esta pode ser a intermediária entre dispositivos conectados.

A questão sobre como implementar a detecção de ataques do tipo *Slow DoS* de maneira rápida e eficiente é uma tarefa desafiadora dada as condições furtivas de ocorrência do ataque. Por ser uma ameaça emergente e não menos preocupante que o DDoS tradicional, o ataque lento de negação de serviço à camada de aplicação, (*Slow DoS*), será o foco deste trabalho.

3 MÉTRICAS PARA A DETECÇÃO *SLOW DOS*

Independentemente da quantidade de máquinas necessárias para empreender um ataque, Jazi et al. (2017) afirma que apenas uma máquina não surte o efeito desejado para o ataque *Slow Send*.

Inicialmente, neste trabalho, foi investigado se a quantidade de endereços IP (q_t), em contato com o servidor alvo, aumentaria subitamente durante um ataque *Slow DoS*, pois desejava-se descobrir se este atributo poderia ser utilizado com as demais métricas para a detecção. Testou-se a observação da variação na quantidade de IP em janelas de tempo distintas (v_t), para verificar se essa variação poderia fornecer evidências do ataque. A percentagem de variação poderia ser determinada pela diferença entre a quantidade de IP na janela de tempo anterior (q_{t-1}) e janela de tempo atual (q_t), dividida pela quantidade na janela de tempo atual, conforme Equação 3.1.

$$v_t = \frac{|q_{t-1} - q_t|}{q_t} \tag{3.1}$$

A métrica foi testada com os dados do CIC-DoS (2017) *Dataset*. O resultado dessa verificação para o primeiro ataque, *Slow Body*, que segundo dados do CIC-DoS (2017), inicia-se a partir do minuto cinquenta e três, não foi satisfatória. O minuto cinquenta e três equivale à janela de número cento e seis, porque o atributo *time* do *dataset* foi dividido em intervalos de tempo ou janelas de tempo de trinta segundos. Observa-se que, a quantidade de endereços IP decresce a partir da janela de tempo cento e seis (minuto cinquenta e três), conforme Figura 3.1.

Logo, um aumento na quantidade de endereços IP não é determinante para a suspeita de um ataque *Slow DoS* em andamento, a despeito de como acontece nos ataques por inundação.

Um súbito aumento na quantidade de endereços IP em *sites* de consultas ou notícias, como por exemplo, em função de divulgação de conteúdo retumbante por parte dos provedores indicam um fenômeno atípico, mas não necessariamente malicioso. Prever os pontos no tempo em que ocorrerão picos de uma demanda é imprevisível. Nesse caso, tal fenômeno surge aleatoriamente em certos momentos e pode consumir toda a capacidade do sistema (MENASCÉ; ALMEIDA, 2003). Nesse sentido, o súbito aumento no número de endereços IP não será o foco do trabalho.

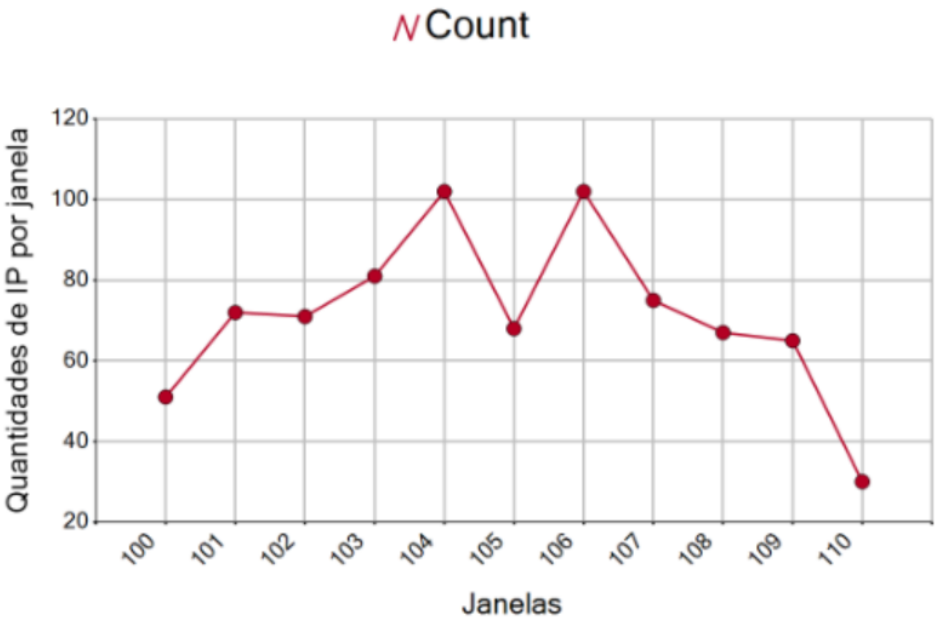


FIG. 3.1: Distribuição da quantidade de IP por janelas de tempo.

A principal particularidade dos ataques lentos à camada de aplicação é a furtividade. São considerados ataques sofisticados, pois agem conforme o protocolo HTTP e estabelecem conexões legítimas, sendo difíceis de serem detectados na camada de rede (XIE; YU, 2009; JAZI et al., 2017). Por exemplo, o ataque *Slowloris* caracteriza-se por picos de tráfego em intervalos regulares (AQIL et al., 2015). Esta peculiaridade, potencializa o ataque, fazendo-o muito difícil de ser detectado, se comparado a um ataque tradicional por inundação.

Por isso, é importante que o sistema de detecção possa estabelecer ou buscar outros padrões característicos de anomalias com base em analogias históricas, para oferecimento de avaliações mais acuradas do tráfego de rede.

Segundo Jazi et al. (2017); Gonzalez et al. (2014), pode-se observar características ou padrões bem definidos nos *traces* maliciosos, conforme listado a seguir:

a) Súbito aumento no número de conexões;

b) Cabeçalhos das requisições HTTP incompletos;

c) Manipulação do Corpo ou conteúdo das requisições HTTP; e

d) Leitura extremamente lenta das respostas do servidor atacado.

As premissas para a aplicação de rede consideradas na metodologia proposta nesse trabalho foram as seguintes:

- Cliente: nome dado ao processo que o cliente utiliza como navegador na Web. É aquele que inicia a comunicação;
- Servidor: nome dado ao processo servidor Web. É o que espera ser contatado para iniciar a sessão;
- Conexão Persistente: o servidor deixa a conexão TCP aberta após enviar resposta. Requisições e respostas subsequentes podem ser enviadas por meio da mesma conexão (KUROSE; ROSS, 2006); e
- Requisições HTTP do cliente bem-sucedidas: quando o servidor entrega ou responde ao cliente o que fôra solicitado, e ambos encerram a comunicação, conforme preconizado pelo protocolo da camada de aplicação.

Conforme já mencionado no capítulo anterior, dentre as diversas técnicas para a mitigação de ataques lentos à camada de aplicação pode-se destacar: limitação de conexões que um servidor pode manipular ou uma limitação de várias conexões simultâneas que um cliente pode ter ao mesmo tempo (GONZALEZ et al., 2014). As métricas desenvolvidas nesse trabalho não destinam-se ao fornecimento de *thresholds* para a mitigação, mas sim à verificação do status verdadeiro dos atributos de rede em um dado momento.

As métricas F.O.R (*Flags, Omnia* e *RTTR*) abordarão as características dos ataques lentos de negação de serviço mediante as premissas para aplicações de rede, com base em hipóteses relacionadas ao comportamento dos atributos do tráfego de rede, que serão descritas a seguir.

3.1 A MÉTRICA *FLAGS*

No estabelecimento das conexões, durante o 3-*way handshake* do protocolo TCP, estão envolvidos os atributos das *Flags* que orientam o estágio das conexões, dentre elas, a *flag* SYN ACK, onde o servidor denota estar em contato com o cliente e a *flag* FIN ACK ao final da conexão.

Conforme elencado no início deste capítulo, uma possível consequência do ataque lento é o súbito aumento no número de conexões. A métrica *Flags* foi projetada para investigar ocorrências das *flags* SYN ACK e FIN ACK das conexões HTTP, presentes no tráfego durante um certo período de tempo.

Por hipótese, as métricas *Flags* deverão atuar na detecção dos aumentos significativos na quantidade de conexões que podem ser observadas com os ataques *Slow DoS*.

Se, no período de tempo considerado, ou seja, dentro da janela de tempo, observa-se o súbito aumento de *flags* SYN ACK, pode-se considerar um comportamento anormal. A anormalidade pode ser confirmada, também, na ausência das *flags* FIN ACK no período considerado, dado que existem mais conexões abertas que fechadas. Para verificar esse tipo de anomalia, estabeleceu-se a métrica *Flags*, para auxiliar a análise do comportamento da rede no que tange ao súbito aumento do número de conexões por período de tempo, conforme a Equação 3.2.

$$Flags = \sum_{i=t_{inicial}}^{t_{final}} SYNACK_i - \sum_{i=t_{inicial}}^{t_{final}} FINACK_i \tag{3.2}$$

A Equação 3.2 consiste na diferença entre o somatório das ocorrências das *flags* SYN ACK e FIN ACK por janela de tempo. Onde i representa o instante t (pacote) por janela de tempo.

Para confirmação da condição de anormalidade, calcula-se a média e respectivos desvios padrão das janelas de tempo mais recentes, para a variável *Flags*, conforme Equações 3.3 e 3.4.

$$\overline{Flags} = \frac{\sum_{i=i-1}^{i-n} Flags_i}{n} \tag{3.3}$$

$$S_{Flags} = \frac{\sum_{i=i-1}^{i-(n-1)} (Flags_i - \overline{Flags})^2}{n-1} \tag{3.4}$$

Para a detecção de anomalia, duas condições deverão ser satisfeitas. A primeira, compara a variável Sum_SYNACK dada pela Equação 3.5.

$$Sum_SYNACK = \sum_{i=t_{inicial}}^{t_{final}} SYNACK_i \tag{3.5}$$

E a segunda, compara o valor da diferença das *flags* da janela atual com relação às janelas mais recentes, segundo o período estipulado pelo previsor média móvel.

Na estatística, em análise de séries temporais, os modelos atribuem peso maior aos dados mais recentes. Então, para a confirmação do estado anormal, ambas as condições citadas deverão atender ao intervalo de confiança estimado pela desigualdade Tchebycheff, que será explicada posteriormente.

O poder da métrica FLAGS concentra-se na investigação das anomalias comuns aos ataques *Slow DoS* (*Slow Headers*, *Slowloris*, *Slowbody*, *R.U.D.Y.* e *Slow Read*).

3.2 A MÉTRICA *OMNIA*

Segundo Gonzalez et al. (2014), a avaliação da capacidade de resposta do servidor é representada pela percentagem de requisições de clientes HTTP deixadas incompletas durante determinado período. Gonzalez et al. (2014) comparou as taxas de responsividade entre diversos servidores Web a fim de medir a resiliência de cada um dos servidores aos ataques *Low Rate*, *Slow Send* e *Slow Read*.

A métrica *Omnia* baseia-se na métrica proposta por Gonzalez et al. (2014), sendo que a *Omnia* analisará as requisições efetivamente incompletas, ou seja, o cabeçalho da requisição não apresenta o segundo CR LF e por isso fica incompleto, fazendo com que a vítima aguarde indefinidamente. Desta forma, a requisição não apresenta a linha em branco ao final do cabeçalho da requisição com o método GET, conforme Figura 3.2.

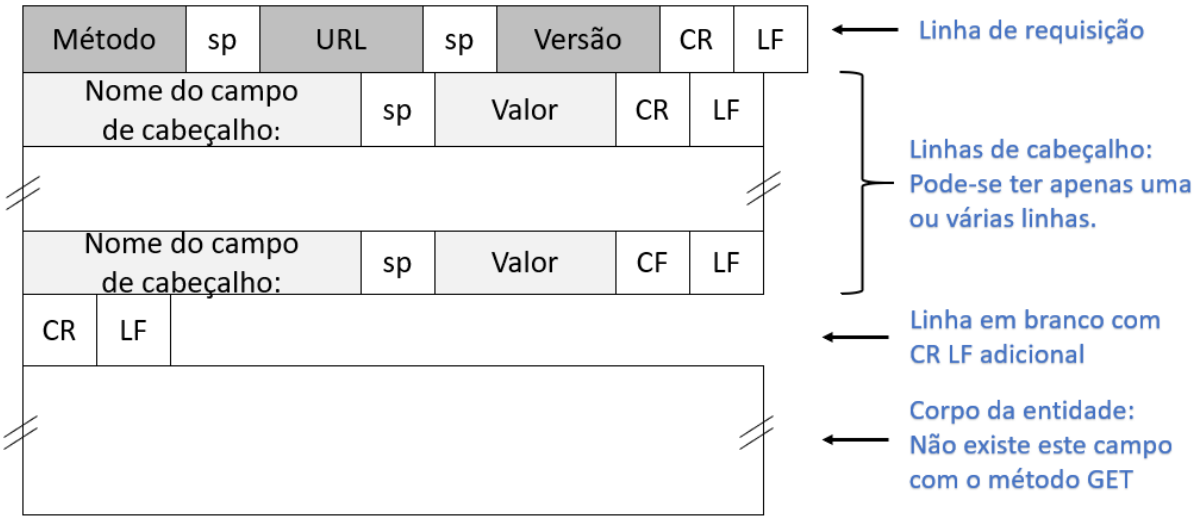


FIG. 3.2: Detalhes de uma requisição HTTP com base em Kurose e Ross (2014).

Segundo Kurose e Ross (2014) a sintaxe da mensagem de requisição do protocolo HTTP diz respeito à como são delineados os campos da mensagem; e a semântica refere-se ao significado da informação nos campos. Tanto a sintaxe quanto a semântica compõem as regras de como um processo envia e responde mensagens e como o processo aguarda dados. Ao não completar o cabeçalho de uma requisição, o atacante beneficia-se da sintática dos campos de cabeçalho da requisição durante o processamento de mensagens com o protocolo HTTP.

A hipótese para a métrica *Omnia* consiste no fato que as anormalidades no somatório de pacotes incompletos por janela de tempo serão detectadas com base no histórico da própria rede.

Esta métrica oferece o cálculo da taxa de requisições incompletas por janela de tempo sendo representa pela Equação 3.6.

$$Omnia = \frac{\sum_{i=t_i}^{t_f} \widetilde{R}_i}{t} \tag{3.6}$$

Onde i é o i-ésimo pacote de dados, t_i é o tempo inicial do intervalo t, t_f é o tempo final do intervalo t, $\widetilde{R}_i$ é definida por requisição incompleta do i-ésimo pacote e t é o intervalo da janela de tempo que pode ser estática ou dinâmica.

Conforme mencionado anteriormente, define-se por requisição incompleta, a requisição HTTP que não apresenta os caracteres CR LF (*Carriage Return Line Feed*) que marcam o final do cabeçalho das requisições com o método GET, conforme Figura 3.3.

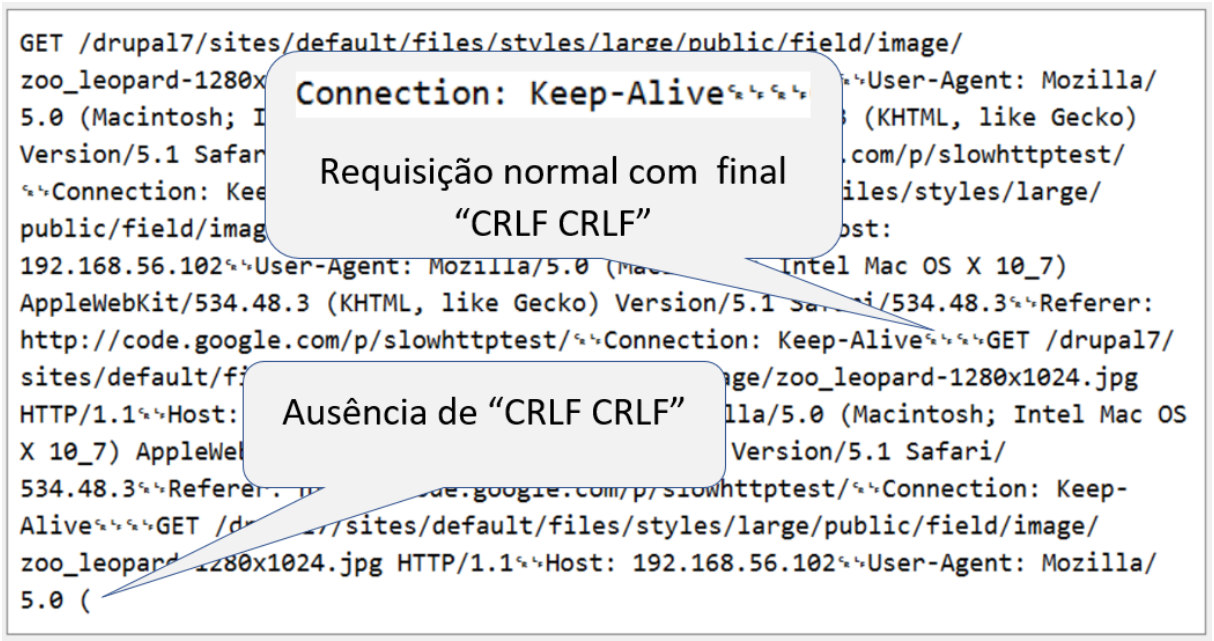


FIG. 3.3: Detalhes de uma requisição HTTP com ausência dos marcadores CR LF CR LF para o último GET. Fonte: CIC-DoS (2017).

De acordo com Damon et al. (2012), a diferença entre uma solicitação GET e uma solicitação POST é que um GET tem apenas um cabeçalho, enquanto um POST tem um cabeçalho e um corpo. Tanto o cabeçalho, quanto o corpo precisam ser terminados com os caracteres CR LF CR LF.

Uma requisição gerada com um formulário não utiliza necessariamente o método POST (GONZALEZ et al., 2014). Ao contrário, formulários HTML costumam empregar o método GET e incluem os dados informados (nos campos do formulário) na URL requisitada. No caso dos formulários HTML, é possível observar os parâmetros próprios desse tipo de solicitação, tais como "?" e "&". Não foram encontrados esses parâmetros

nas solicitações com o método GET no exemplo demonstrado pela Figura 3.3.

A Figura 3.3 poderia demonstrar um ataque *Slowloris* ou *Slow Body*. No entanto, pelo exposto anteriormente, a possibilidade de ser um ataque *Slow Body* foi desconsiderada.

Tal argumentação deve-se ao fato que, o ataque representado na Figura 3.3 pertence à categoria *Slow Read* segundo dados fornecidos pelo CIC-DoS (2017).

O poder da métrica *Omnia* concentra-se na investigação das anomalias comuns aos ataques *Slow Headers*, *Slowloris* e *Slow Body*, que utilizam a técnica do envio de solicitações de forma lenta, utilizando o artifício das requisições incompletas.

3.3 A MÉTRICA *RTTR*

O RTT (*Round Trip Time*) é o tempo decorrido entre o envio de um pacote e o recebimento do respectivo reconhecimento (KUROSE; ROSS, 2006). Os atrasos percebidos por este atributo da camada de transporte (RTT) podem ser causados pelo envio dos cabeçalhos das requisições HTTP incompletos ao servidor e pela manipulação do conteúdo dos atributos do tráfego de rede, tais como: janela de recepção, *content-length*, entre outros, durante o envio dessas mensagens ao servidor e vice-versa.

Por exemplo, se um atacante envia diversos pacotes incompletos, o servidor irá respondê-lo, mas o cliente mal-intencionado demorará mais tempo que o normal para enviar o reconhecimento (ACK) às mensagens do servidor, pois o que o cliente realmente deseja é manter a atenção do servidor pelo maior tempo possível.

Em outra situação, caso a janela de recepção seja manipulada, o servidor, por não ter um limite para a duração das conexões com o cliente, enviará o solicitado em pacotes de tamanho mínimo. Assim, as mensagens de resposta terão tamanho extremamente reduzido e com isso o cliente ilegítimo vai conseguindo manter a atenção do servidor, pois demora a dar o reconhecimento de tudo o que for enviado pelo servidor.

A concretização do RTT lento ocorrerá no servidor. Assim, as mensagens de resposta terão tamanho extremamente reduzido, e dessa forma o objetivo do atacante, que é a manutenção da conexão com o servidor pelo maior período possível, é atingido.

Portanto, a métrica RTTR (RTT Real) buscará por esses tipos de anormalidade. Assim, a hipótese associada à métrica RTTR, é que a média de RTTR por janela, que se concretizam no servidor, levarão à detecção de anormalidades dos ataques *Slow DoS*.

Para averiguação dos comportamentos do RTT fora do padrão, assume-se que o servidor será a fonte e o cliente o destino. Assim, o tempo começa no servidor e termina no recebimento do reconhecimento para o pacote emitido ao servidor. Confirmando que, o

valor RTT se concretizará no servidor, conforme Fig.3.4.

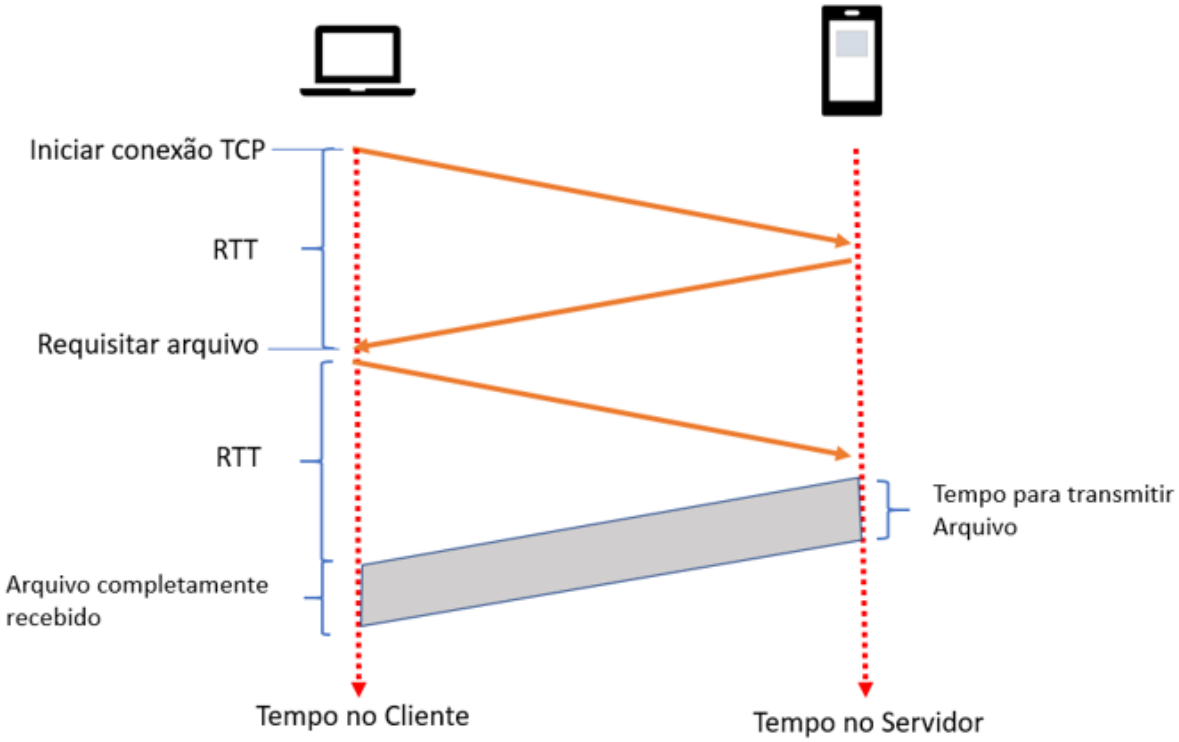


FIG. 3.4: Tempo de ida e volta de pacotes com base em Kurose e Ross (2006)

A métrica RTTR foi projetada em função do RTT Real e não o valor de RTT suavizado, que é utilizado para gerenciamento da temporização de transmissão. Para evitar que o remetente sature o *buffer* do destinatário rapidamente, o protocolo TCP utiliza o controle de fluxo, fazendo com que o remetente mantenha a variável janela de recepção sob qualquer suspeita (KUROSE; ROSS, 2006).

Calcula-se a métrica RTTR pelo somatório dos valores de RTT por janela de tempo de acordo com Equação 3.7 .

$$RTTR = \sum_{i=t_{inicial}}^{t_{final}} \mathrm{RTT}_i \tag{3.7}$$

O poder da métrica RTTR concentra-se na investigação das anomalias comuns aos ataques *Slow* DoS (*Slow Headers*, *Slowloris*, *Slowbody*, $R.U.D.Y$ e *Slow Read*).

Apresenta-se na Tabela 3.1 , um resumo das características mensuradas pelas métricas F.O.R.

As características mensuradas pelas métricas, propostas nesse trabalho, buscam detectar as anormalidades comuns aos ataques da família *Slow DoS*, que segundo Gonzalez et al. (2014) exploram as propriedades comuns ao protocolo da camada de aplicação, prin-

TAB. 3.1: Métricas F.O.R para a Detecção de Anomalias *Slow DoS*

Métrica	Características Mensuradas	Ataque Investigado
Flags	Discrepância na proporção das *flags* de estabelecimento e encerramento de conexão do protocolo TCP.	*Slow Headers, Slowloris, Slow Body, R.U.D.Y. e Slow Read*
Omnia	Requisições incompletas no protocolo HTTP.	*Slow Headers e Slowloris*
RTTR	Distorção percebidas no servidor com relação ao tempo de resposta durante as transações com o protocolo HTTP.	*Slow Headers, Slowloris, Slow Body, R.U.D.Y. e Slow Read*

cipalmente, os parâmetros de tempo no lado do servidor, fazendo com que este reserve recursos até o término da conexão.

4 METODOLOGIA PARA O SISTEMA DE DETECÇÃO

Conforme mencionado no capítulo anterior, os sistemas de detecção baseiam-se na assinatura dos ataques ou buscam padrões de comportamentos anômalos no tráfego de rede, principalmente, para realimentar os métodos empregados na mitigação dos problemas ocasionados pelos ataques à rede.

Outra classificação importante em um sistema de detecção é o local onde o sistema deverá atuar: se próximo à fonte, próximo à vítima ou em roteadores intermediários. Segundo a taxonomia proposta por Gupta e Badve (2017), o tráfego é mais disperso próximo à fonte, e portanto, torna-se muito difícil encontrar pacotes maliciosos, pois parecem pacotes inofensivos.

O sistema proposto detecta anomalias causadas pelos ataques *Slow DoS*, com base nas métricas concebidas no presente trabalho, derivadas das heurísticas descritas no capítulo anterior. O mecanismo desse sistema baseia-se na detecção de anomalias com base no comportamento dos atributos de rede e o local onde o mecanismo atua é próximo à vítima.

A arquitetura do sistema para detecção de anomalias *Slow DoS*, é apresentada na Figura 4.1.

Cada janela de tempo será estimada dinamicamente pela lógica *fuzzy*, antes que o tráfego seja encaminhado à fase de processamento pelas métricas.

A estratégia da metodologia proposta consiste na averiguação dos resultados das métricas em intervalos ou janelas de tempo definidas previamente.

Tal estratégia é motivada pelos métodos tradicionais de votação para combinação de classificadores, que por sua vez, baseiam-se na teoria do aprendizado bayesiano. Nessa teoria, ao invés de utilizar um único modelo de decisão, preconiza-se que devem ser utilizados todos os modelos aceitáveis no espaço da hipótese (FACELI et al., 2015). Na presente metodologia, o espaço hipótese representa o resultado das métricas.

As métricas compõem um comitê de votação com peso uniforme e processam seus resultados com execução paralela. Assim, todas as métricas participam do processo de verificação de anomalia e fornecem o seu resultado ao decisor, após o processo de classificação, onde ocorrerá a rotulação do tráfego em normal ou anormal, segundo a desigualdade Tchebycheff.

Assim, dentro do espaço hipótese, dado que uma das métricas detectou anormalidade,

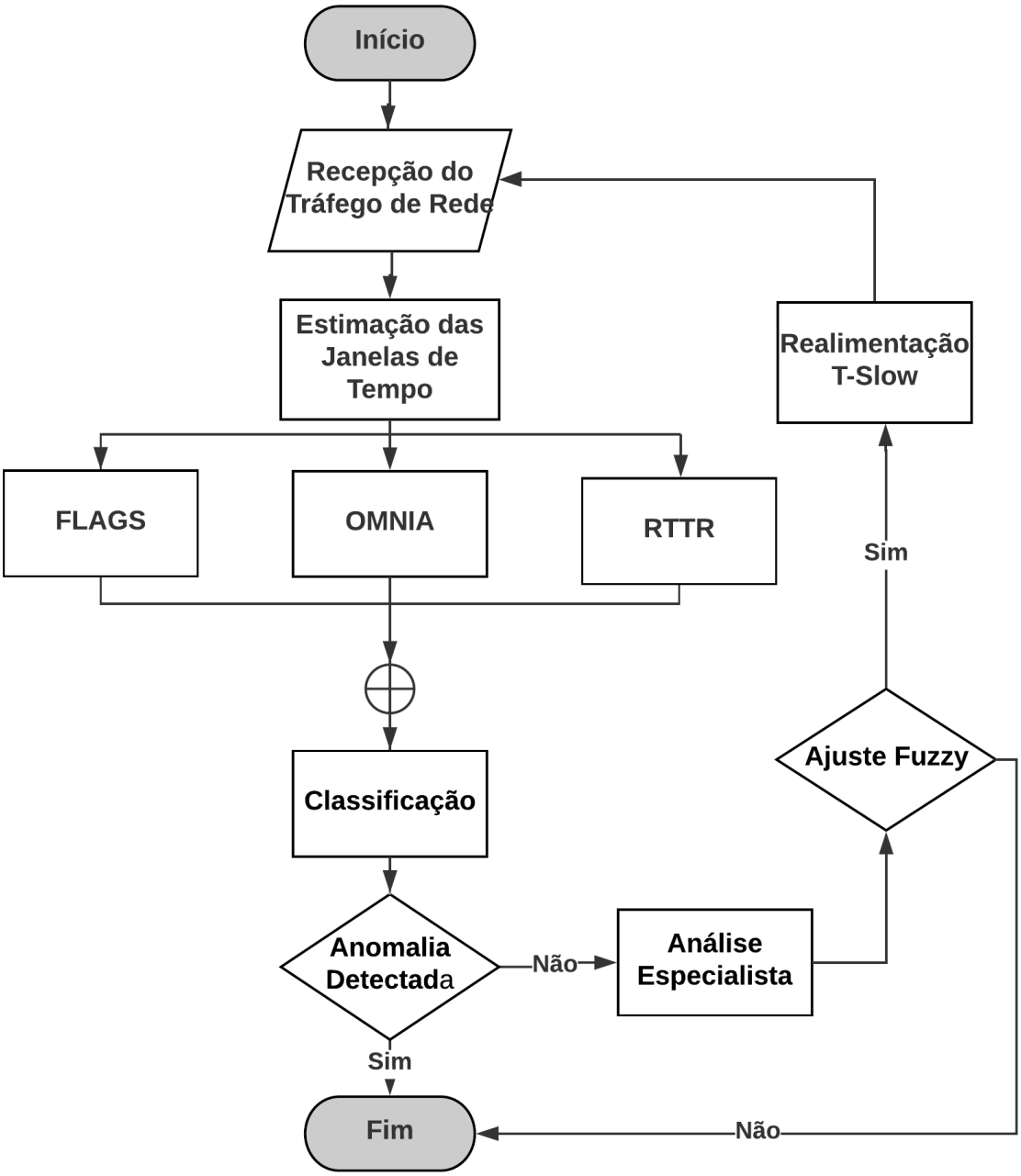


FIG. 4.1: Arquitetura para detecção de anomalias dos ataques *Slow DoS*.

o classificador rotula o período analisado como anormal.

Para a janela de tempo analisada, o decisor emitirá sua conclusão final do *T-Slow*, caso pelo menos uma das métricas forneça o resultado positivo para ataque. Assim, caso a anomalia seja detectada, o processo finaliza. Caso a anomalia não seja detectada, a informação do tráfego poderá ser submetida à análise do Especialista, que verificará, por exemplo, a ocorrência de falsos negativos. Nesse caso, pode-se fazer necessário o ajuste da

lógica *fuzzy*, tais como mudanças nos valores do discurso real das funções de mapeamento. Caso o ajuste não seja necessário, o processo termina. Se o ajuste for necessário, haverá o processo de realimentação do algoritmo *T-Slow*, já com as novas estimativas das janelas de tempo.

4.1 ESTIMAÇÃO DO TAMANHO DAS JANELAS

Dada a furtividade dos ataques *Slow DoS*, dispor de uma janela de tempo adequada, pode contribuir na detecção das anomalias com baixo número de falsos positivos e falsos negativos. Essas taxas determinam a acurácia total do sistema de detecção.

O atributo de rede *time*, presente no *dataset* empregado, foi utilizado no cálculo das janelas de tempo. Ampliando a experiência, além das janelas determinadas pela lógica *fuzzy*, as métricas também foram submetidas às janelas com intervalo de tempo fixo.

Nas subseções seguintes serão apresentadas dois tipos de abordagens. Primeiramente, será apresentado o método de estimação com janelas estáticas ou fixas e posteriormente, a estimação das janelas dinâmicas com a lógica *fuzzy*.

4.1.1 JANELAS DE TEMPO ESTÁTICAS

Um dos parâmetros de avaliação de performance de um sistema de detecção é o atraso com que o sistema de detecção pode alertar sobre a ocorrência do ataque ou *timeliness*. Essa avaliação está ilustrada na Figura 4.2.

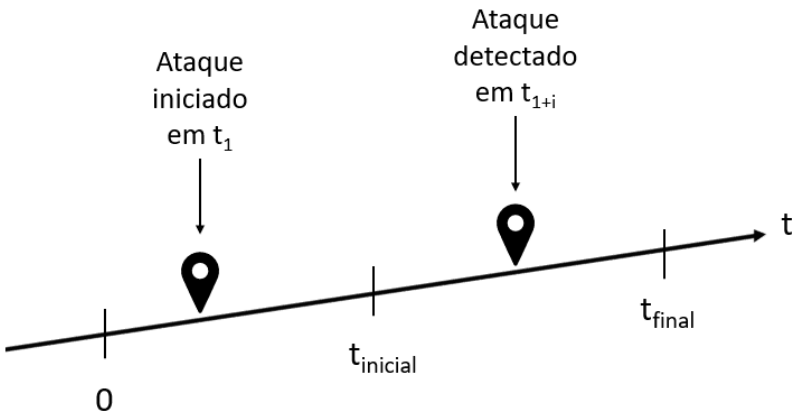


FIG. 4.2: Timeliness para detecção de anomalias.

Com o *dataset* utilizado na metodologia, não foi possível constatar o parâmetro *timeliness*, pois não foi disponibilizado o momento exato do início do ataque.

Com base em experimentos realizados em outros trabalhos de detecção (JAZI et al., 2017; HIRAKAWA et al., 2016), adotou-se a janela de tempo fixa com 30 segundos. A

captura de tráfego do CIC-DoS (2017) *dataset* teve início à zero hora, um minuto, seis segundos e 29.000 milissegundos do dia 12 de junho de 2010, onde o cronômetro time marca 0.00000 segundos, conforme Tabela 4.1 . O atributo *time* desse tráfego foi dividido em intervalos menores de trinta (30) segundos, dando origem ao atributo janela de tempo. Neste caso, o tamanho da janela é diretamente proporcional ao tráfego.

TAB. 4.1: Período de Captura de Tráfego no *Dataset* com base no CIC-DoS (2017).

Atributos	Início	Fim
Time (segundos)	0.000000	86399.955436
Data-Hora	12-JUN-2010 00:01:06:029000	13-JUN-2010 00:01:05:984436

Assim, o processo de estimação das janelas de tempo da Figura 4.1, obteria uma distribuição de intervalos de tempo, conforme representação da Tabela 4.2.

TAB. 4.2: Distribuição dos Intervalos para as Janelas de Tempo de 30 segundos com os dados do CIC-DoS (2017).

Janela	Intervalo
0	$0 \vdash 29$
1	$30 \vdash 59$
2	$60 \vdash 89$
.	.
.	.
.	.
2880	$86.400 \vdash 86.429$

Os ataques *Slow DoS* aproveitam-se da leniência por parte de alguns servidores *Web* (DAMON et al., 2012; HELALAT, 2017) com relação aos atrasos na rede. Por este motivo, a detecção deve atentar para as mudanças de comportamento na rede. O resultado da detecção retornará a janela de tempo, cujos valores das métricas F.O.R. foram anormais. Os resultados obtidos estão no Capítulo 5 sobre o estudo de caso com o algoritmo T-*Slow*.

4.1.2 JANELAS DE TEMPO DINÂMICAS COM A LÓGICA FUZZY

Na literatura, a divisão do tráfego de rede para análise de seus atributos, costuma ser mediante das janelas estáticas testadas empiricamente como em Jazi et al. (2017); Tripathi e Hubballi (2018). São raras as discussões sobre a estimação das janelas dinâmicas, ou seja, as janelas de tempo que se adaptam, conforme o comportamento revelado pelos atributos do tráfego de rede, como proposto por dos Santos (2009).

O presente experimento, com a estimação do tamanho das janelas dinâmicas, utilizou a lógica *fuzzy* e teve o propósito de apresentar análises mais próximas da realidade para a estimação da variação no tamanho das janelas de tempo, conforme a metodologia de detecção. Estas análises referem-se à determinação do tamanho adequado da janela de tempo para análise de certos atributos do tráfego de rede, o que é muito difícil de precisar.

A rede neural não foi escolhida para essa tarefa, porque segundo Faceli et al. (2015), a desvantagem desta técnica, é que a extração de conhecimento de uma rede neural artificial é uma tarefa que exige recursos e esforços adicionais, e que se não for bem justificada, pode apresentar efeitos negativos.

A lógica *fuzzy* é uma extensão de um conjunto clássico, onde a pertinência de um elemento ao conjunto é interpretada como sendo uma situação intermediária entre a total pertinência e a não pertinência (JANTZEN, 1998; DOS SANTOS, 2009).

Um conjunto *fuzzy* provê uma forma matemática de representar a imprecisão e incerteza nos sistemas humanos (ROSS, 2009). Assim, a determinação da variação do tamanho da janela de tempo será precisado pela lógica *fuzzy*.

Dessa forma, o processo de estimação das janelas de tempo da Figura 4.1, obedecerá a lógica *fuzzy* e consequentemente, o tamanho das janelas será variável ao longo do tempo.

As Variáveis linguísticas para os conjuntos *Fuzzy* são:

- Max_RTT: RTT máximo por janela; e
- Avg_Est_RTT: média para o RTT estimado por janela.

Define-se por RTT Máximo, o maior RTT amostrado (*sample RTT*) por janela de tempo. O RTT amostrado é também chamado RTT real e relaciona-se a cada reconhecimento ACK (*acknowledgment*), conforme a RFC 6298.

Assim, o RTT máximo (Max_RTT) não é um RTT estimado(RTT_Est), mas sim um RTT real.

Devido à suscetibilidade de variações atípicas em decorrência do congestionamento nos roteadores e das variações de carga nos sistemas finais, o protocolo TCP utiliza o RTT estimado (suavizado) para fins de cálculo de congestionamento de rede (KUROSE; ROSS, 2006), sendo calculado pela Equação 4.1.

$$Est_RTT = (1 - \alpha) .Est_RTT + \alpha.SampleRTT \tag{4.1}$$

Com a média dos valores estimados de RTT pode-se também identificar grandes variações de valores.

Deseja-se que a lógica *fuzzy* seja capaz de processar as variações atípicas desses valores e conduza a melhor decisão para o tamanho das janelas de tempo.

As partições *fuzzy* das variáveis linguísticas, tanto para a variável Max_RTT quanto para a variável Avg_RTT_Est, são as seguintes:

- $\mu_{\mathbf{P}}(\mathbf{x})$: Pequeno;
- $\mu_{\mathbf{M}}(\mathbf{x})$: Médio e;
- $\mu_{\mathbf{G}}(\mathbf{x})$: Grande.

As funções de pertinência possuem em suas abscissas, valores coerentes com aqueles existentes no processo real (DOS SANTOS, 2009).

Pela teoria clássica dos conjuntos, pode-se descrever os objetos de natureza contínua da seguinte forma: $U : \{x \in \mathcal{R}_+\}$.

A teoria dos conjuntos *fuzzy* pode ser vista como uma extensão da teoria clássica de conjuntos e foi criada para tratar graus de pertinência intermediários entre a pertinência total e a não-pertinência de elementos de um universo de discurso com relação a um dado conjunto (REZENDE, 2003). Assim, algebricamente, a função é mapeada conforme equação 4.2.

$$\mu_y(x) : U \rightarrow [0, 1] \tag{4.2}$$

Para exemplificar o universo do discurso real da variável Max_RTT, a Figura 4.3 mostra o gráfico onde a abscissa mostra intervalos de tempo de um segundo e, na ordenada, consta o valor do RTT máximo correspondente. Observa-se que nesse intervalo, devido a ocorrência do ataque *Slowloris*, o valor do RTT_máximo foi igual a 3 segundos. De forma semelhante, as Figura 4.4, 4.5 e 4.6, também mostram aumentos nos valores da variável Max_RTT para os ataques *Slow Headers*, *Slow Body* e *Slow Read*, respectivamente. Os maiores valores para a variável podem ser observados na Figura 4.7, onde o valor da ordenada RTT_máximo mede mais que trinta segundos.

Identificada a variável linguística Max_RTT, com os respectivos valores pertinentes ao discurso real, o próximo passo é o mapeamento algébrico desta variável. O mapeamento consiste na caracterização dos valores do discurso real em um conjunto *fuzzy*. Essa representação dependerá da natureza da variável linguística.

A variável Max_RTT possui natureza contínua e portanto, será representada pelos graus de pertinência, conforme Equações 4.3, 4.4 e 4.5, para os dados do CIC-DoS (2017) *Dataset*.

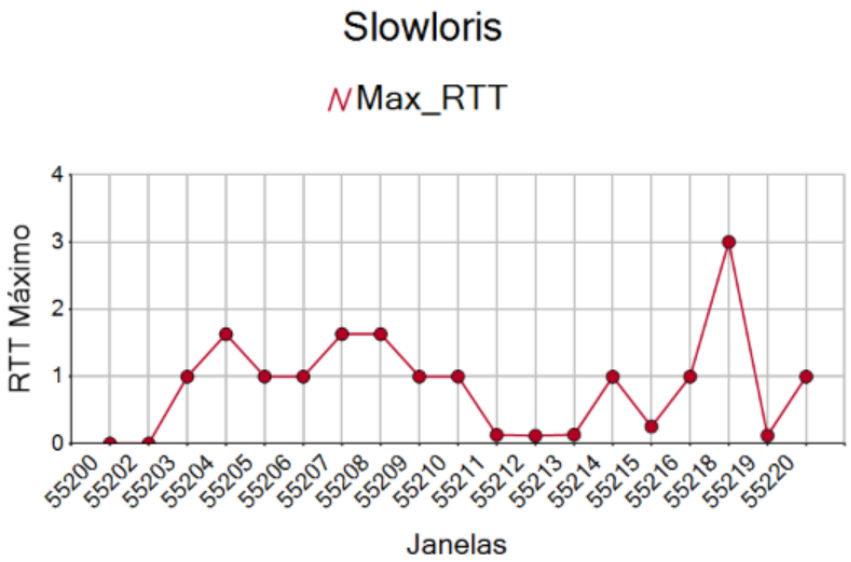


FIG. 4.3: Slowloris - Valores reais do discurso para a variável Max_RTT com base nos dados do CIC-DoS (2017) *Dataset*.

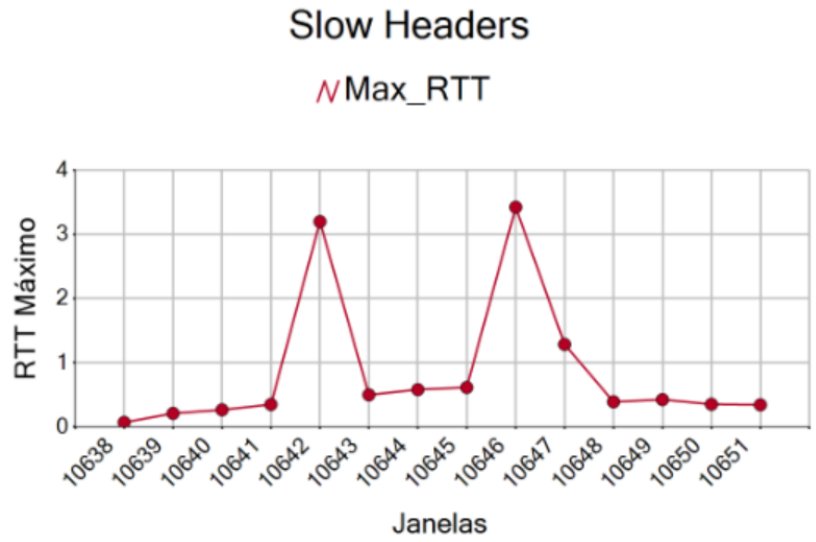


FIG. 4.4: Slow Headers - Valores reais do discurso para a variável Max_RTT com base nos dados do CIC-DoS (2017) *Dataset*.

Utilizando-se outro *dataset*, os valores das variáveis linguísticas poderão ser reajustados no processo de realimentação, conforme metodologia proposta nesse trabalho.

Quando o universo do discurso é contínuo ou possui uma quantidade grande de elementos discretos, o gráfico de sua função de pertinência é chamado de diagrama de Hassi-Euler (H-E) (REZENDE, 2003).

Os valores do discurso real serão utilizados para construir as equações das funções de cada um dos diagramas Hassi-Euler. Foi utilizada a equação reduzida da reta, dada pela Equação $y = ax + b$.

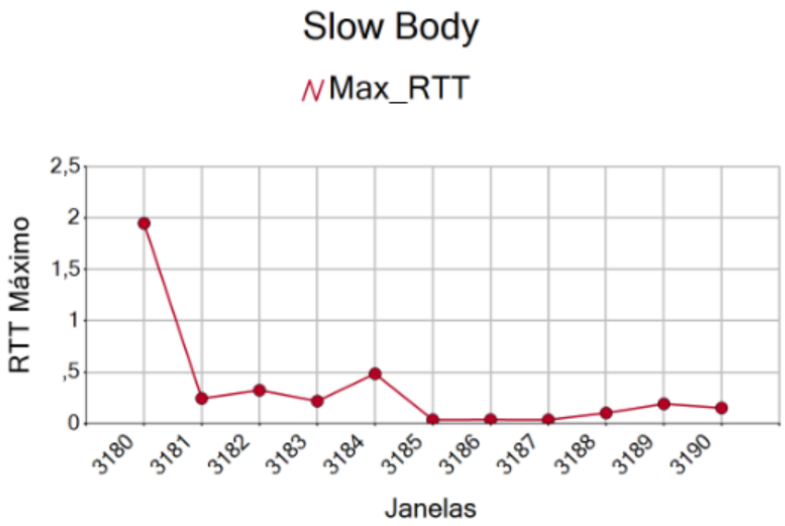


FIG. 4.5: Slow Body - Valores reais do discurso para a variável Max_RTT com base nos dados do CIC-DoS (2017) *Dataset*.

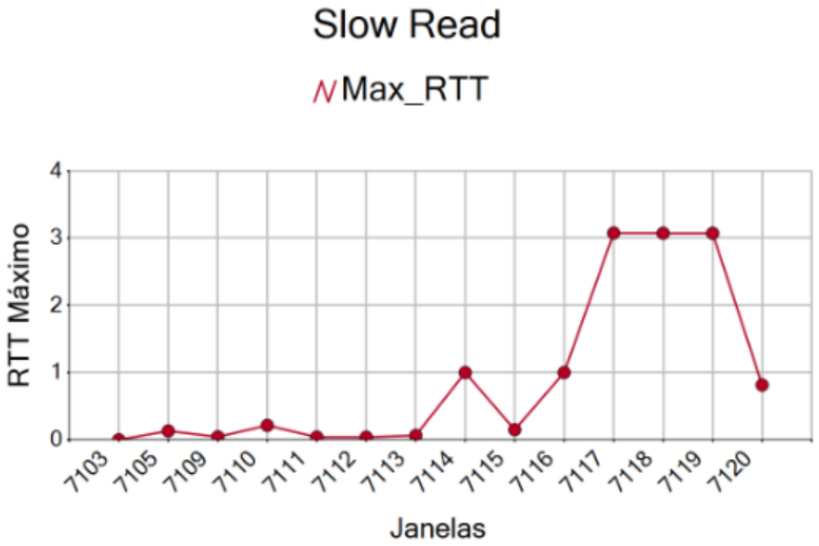


FIG. 4.6: Slow Read - Valores reais do discurso para a variável Max_RTT com base nos dados do CIC-DoS (2017) *Dataset*.

Assim, com o mapeamento algébrico da variável Max_RTT, obteve-se os conjuntos *fuzzy*: $\mu_{\mathbf{P}}(\mathbf{x})$; $\mu_{\mathbf{M}}(\mathbf{x})$ e $\mu_{\mathbf{G}}(\mathbf{x})$, onde o valor zero (0) representa a total falta de pertinência do valor da variável linguística ao conjunto *fuzzy* $\mu(\mathbf{x})$ e; o valor um (1) representa a total pertinência do valor da variável linguística ao conjunto *fuzzy* $\mu(\mathbf{x})$.

$$\mu_{\mathbf{P}}(\mathbf{x}) = \begin{cases} 1 & se \;\; x < 0,5 \\ -2x + 2 & se \; 0,5 \leq x \;\; \leq 1,0 \\ 0 & se \; x > 1,0 \end{cases} \tag{4.3}$$

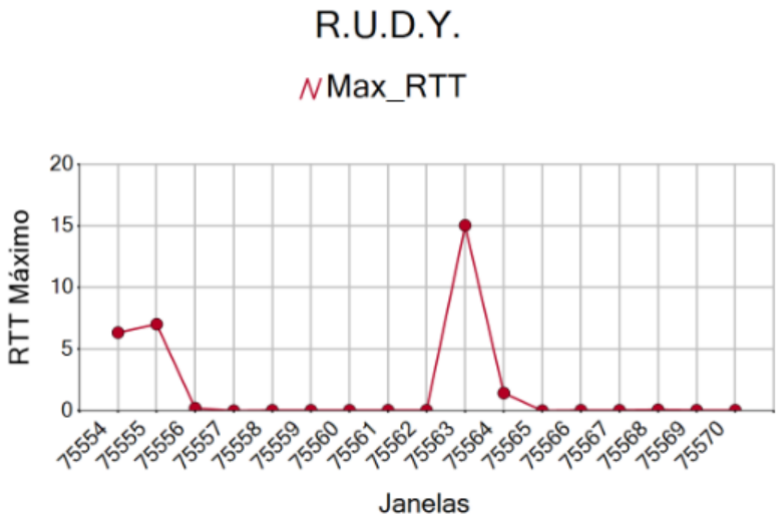


FIG. 4.7: R.U.D.Y - Valores reais do discurso para a variável Max_RTT com base nos dados do CIC-DoS (2017) *Dataset*.

$$\mu_{\mathbf{M}}(\mathbf{x}) = \begin{cases} 0 & se \;\; x < 0,5 \\ -2x \;\; -1 & se \; 0,5 \leq x \;\; \leq 1,0 \\ -2x + 3 & se \; 1,0 < x \leq 1,5 \end{cases} \tag{4.4}$$

$$\mu_{\mathbf{G}}(\mathbf{x}) = \begin{cases} 0 & se \;\; x < 1,0 \\ 2x - 2 & se \; 1,0 \leq x \;\; \leq 1,5 \\ 1 & se \; x > 1,5 \end{cases} \tag{4.5}$$

Para a variável linguística Avg_RTT_Est, os valores do universo do discurso real, podem ser exemplificados pela Figura 4.8, onde na abscissa estão os intervalos de tempo em segundos e na ordenada, os valores para a média do RTT estimado correspondente.

Para essa variável linguística, observa-se que os valores médios são medidos na casa dos milissegundos, conforme Figuras 4.9 e 4.10. A variável apresentou maior valor para média do RTT estimado (acima de 4 segundos) com o ataque *R.U.D.Y.*, conforme Figura 4.11.

A representação algébrica dos conjuntos *fuzzy*: $\mu_{\mathbf{P}}(\mathbf{x})$, $\mu_{\mathbf{M}}(\mathbf{x})$ e $\mu_{\mathbf{G}}(\mathbf{x})$, com os respectivas graus de pertinência para a variável média de Est_RTT (Avg_Est_RTT) estão descritos nas Equações 4.6, 4.7 e 4.8.

$$\mu_{\mathbf{P}}(\mathbf{x}) = \begin{cases} 1 & se \;\; x < 0,05 \\ -20x + 2 & se \; 0,05 \leq x \;\; \leq 0,1 \\ 0 & se \; x > 0,10 \end{cases} \tag{4.6}$$

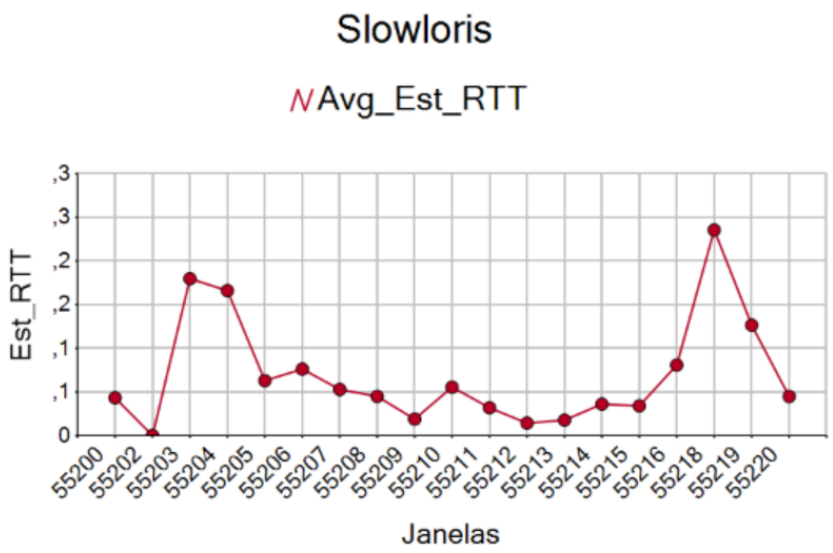


FIG. 4.8: Slowloris - Valores reais do discurso para a variável Est_RTT com base nos dados do CIC-DoS (2017) *Dataset*.

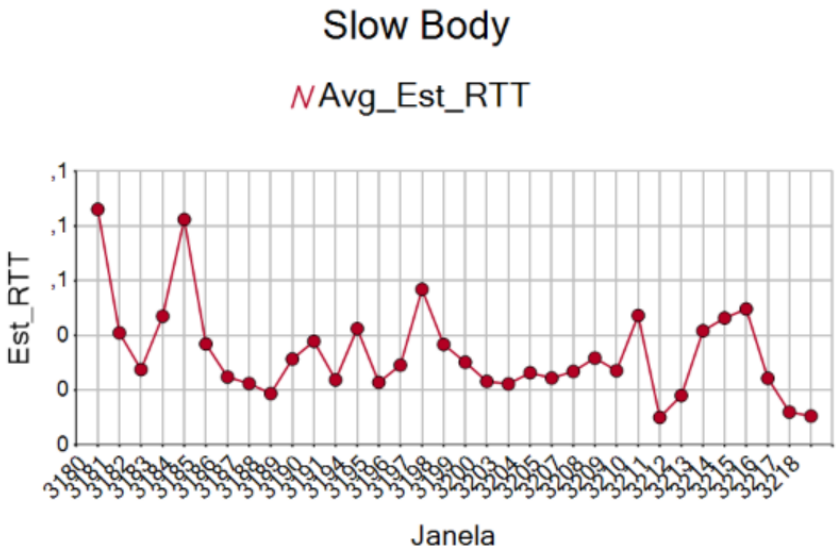


FIG. 4.9: Slow Body - Valores reais do discurso para a variável Est_RTT com base nos dados do CIC-DoS (2017) *Dataset*.

$$\mu_{\mathbf{M}}(\mathbf{x}) = \begin{cases} 0 & se \quad x < 0,05 \\ 20x - 1 & se\ 0,05 \leq x \leq 0,10 \\ -20x + 3 & se\ 0,10 < x \leq 0,15 \end{cases} \tag{4.7}$$

$$\mu_{\mathbf{G}}(\mathbf{x}) = \begin{cases} 0 & se \quad x < 0,10 \\ 20x - 2 & se\ 0,10 \leq x \leq 0,15 \\ 1 & se\ x > 0,15 \end{cases} \tag{4.8}$$

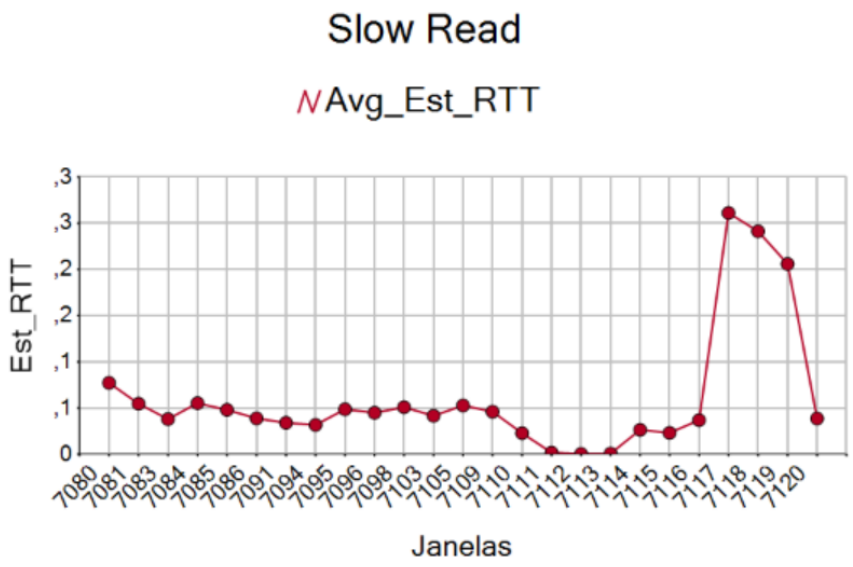


FIG. 4.10: Slow Read - Valores reais do discurso para a variável Est_RTT com base nos dados do CIC-DoS (2017) *Dataset*.

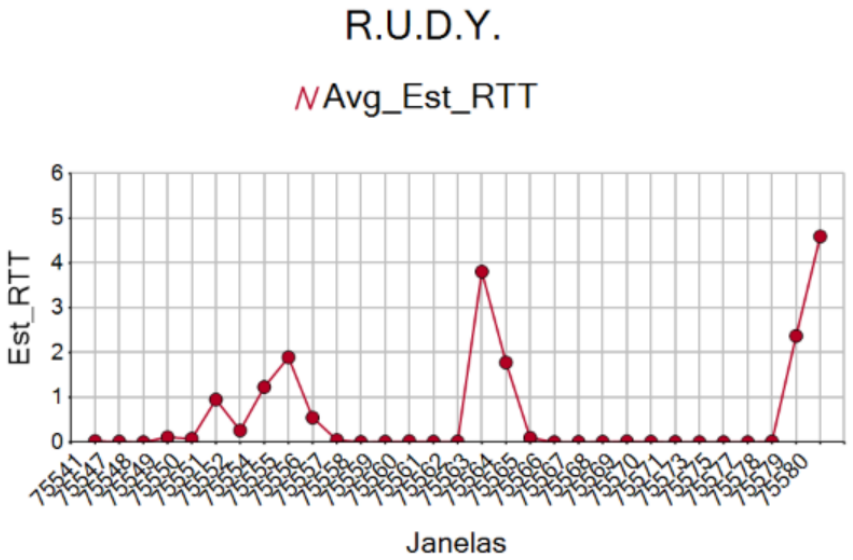


FIG. 4.11: R.U.D.Y - Valores reais do discurso para a variável Est_RTT com base nos dados do CIC-DoS (2017) *Dataset*.

Conforme mencionado anteriormente, o valor estimado é utilizado para fins de cálculo de congestionamento e o valor do RTT Máximo é o valor real do RTT para cada reconhecimento (ACK) que o servidor recebe.

É possível constatar entre os valores de ambas variáveis linguísticas, que o valor do RTT suavizado apresenta valores de RTT menores que o valor de Max_RTT. Isto se deve ao fato que o Max_RTT representa o RTT real. Essas variáveis linguísticas foram escolhidas, porque acredita-se que, as essas duas instâncias do atributo RTT, juntas,

possam representar ou corroborar a situação de lentidão do tráfego.

Os gráficos das partições *fuzzy* para Max_RTT e Avg_Est_RTT são demostrados pelas Figuras 4.12 e 4.13, respectivamente, mediante o diagrama de Hassi-Euler.

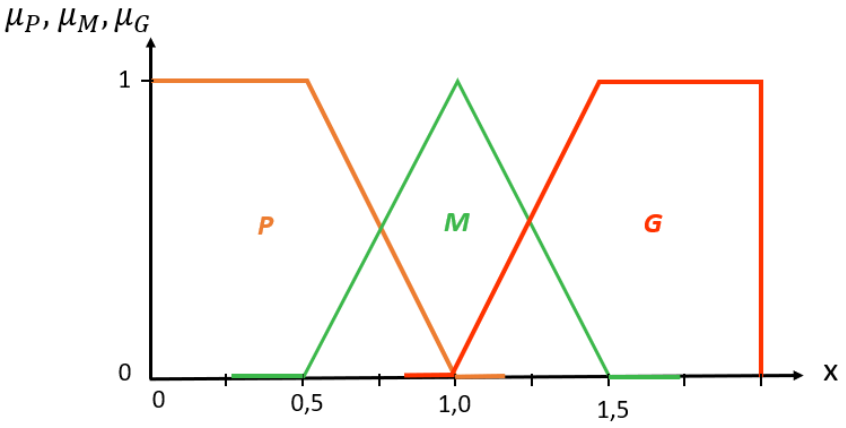


FIG. 4.12: Variável Max_RTT - Partição *Fuzzy*.

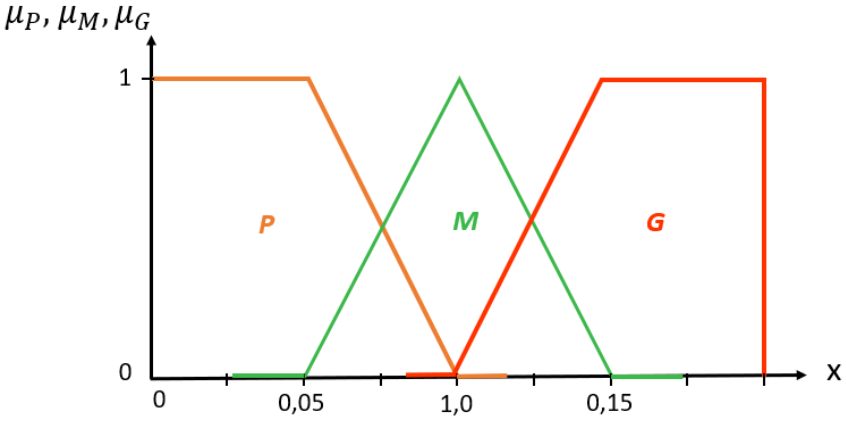


FIG. 4.13: Variável Est_RTT - Partição *Fuzzy*.

As Figuras 4.12 e 4.13 traduzem conjuntos de possíveis soluções para o problema da estimação do tamanho da janela de tempo, classificando as variáveis em termos de conceitos qualitativos (pequeno, médio e grande). Segundo Rezende (2003), a conversão de relações essencialmente gerais e imprecisas, obtidas em um primeiro momento, em algoritmos operacionais mais detalhados numa segunda fase é utilizada pelo ser humano que, primeiro, tenta estruturar o conhecimento sobre este problema em conceitos gerais e depois, as relações essenciais entre esses conceitos.

Os métodos mais utilizados na literatura de inferência dedutiva para sistemas *fuzzy* são: o modelo de Mamdani , modelo Takagi-Sugeno-Kang e modelo Tsukamoto (ROSS, 2009). O modelo para processamento das regras de produção escolhido foi o de Mamdani

(MAMDANI; ASSILIAN, 1975), porque este modelo pode ser utilizado com qualquer número de antecedentes (entradas) e qualquer número de consequentes (saídas). Nesse modelo, as regras de produção possuem relações *fuzzy* tanto em seus antecedentes como em seus consequentes (REZENDE, 2003), sendo representadas da seguinte forma:

$$if \ < antecedente > \rightarrow < consequente > \tag{4.9}$$

Com os antecedentes, o mecanismo de inferência determinará o processamento do consequente da regra *fuzzy*, mediante a seguinte semântica:

$$if \quad x_1 = A_i \quad \wedge \quad x_2 \quad = A_j \quad \rightarrow \quad y_1 = B_m \tag{4.10}$$

O modelo utilizado nesse trabalho baseia-se no modelo utilizado por dos Santos (2009), para monitorar e alterar o tamanho das janelas de tempo.

Obteve-se os conjuntos fuzzy $\mu_{NP}(y)$, $\mu_{NG}(y)$, $\mu_{Z}(y)$, $\mu_{PP}(y)$, e $\mu_{PG}(y)$, com os respectivos graus de pertinência para VJ, conforme equações 4.11, 4.12, 4.13, 4.14 e 4.15.

$$\mu_{\mathbf{NG}}(\mathbf{x}) = \begin{cases} 0 & se \quad -\frac{J}{4} \leq x < -\frac{3J}{4} \\ \frac{4}{J}x + 3 & se \quad -\frac{J}{2} < x \leq -\frac{J}{4} \\ -\frac{4}{J}x - 1 & se \quad -\frac{J}{2} < x < -\frac{J}{4} \end{cases} \tag{4.11}$$

$$\mu_{\mathbf{N}P}(\mathbf{x}) = \begin{cases} 0 & se \quad 0 \leq x < -\frac{J}{2} \\ \frac{4}{J}x + 2 & se \quad -\frac{J}{2} < x \leq -\frac{J}{4} \\ -\frac{4}{J}x & se \quad -\frac{J}{4} < x < 0 \end{cases} \tag{4.12}$$

$$\mu_{Z}(\mathbf{x}) = \begin{cases} 0 & se \quad \frac{J}{4} \leq x < -\frac{J}{4} \\ \frac{4}{J}x + 1 & se \quad -\frac{J}{4} \leq x \leq 0 \\ -\frac{4}{J}x + 1 & se \quad 0 < x < \frac{J}{4} \end{cases} \tag{4.13}$$

$$\mu_{PP}(\mathbf{x}) = \begin{cases} 0 & se \quad \frac{J}{2} \leq x < 0 \\ \frac{4}{J}x & se \quad 0 \leq x \leq \frac{J}{4} \\ -\frac{4}{J}x + 2 & se \quad \frac{J}{4} < x < -\frac{J}{2} \end{cases} \tag{4.14}$$

$$\mu_{PG}(\mathbf{x}) = \begin{cases} 0 & se \quad \frac{3J}{4} \leq x < \frac{J}{4} \\ \frac{4}{J}x + 3 & se \quad \frac{J}{4} \leq x \leq \frac{J}{2} \\ -\frac{4}{J}x - 1 & se \quad \frac{J}{2} < x < \frac{3J}{4} \end{cases} \tag{4.15}$$

O gráfico da partição para VJ é representado pela Figura 4.14.

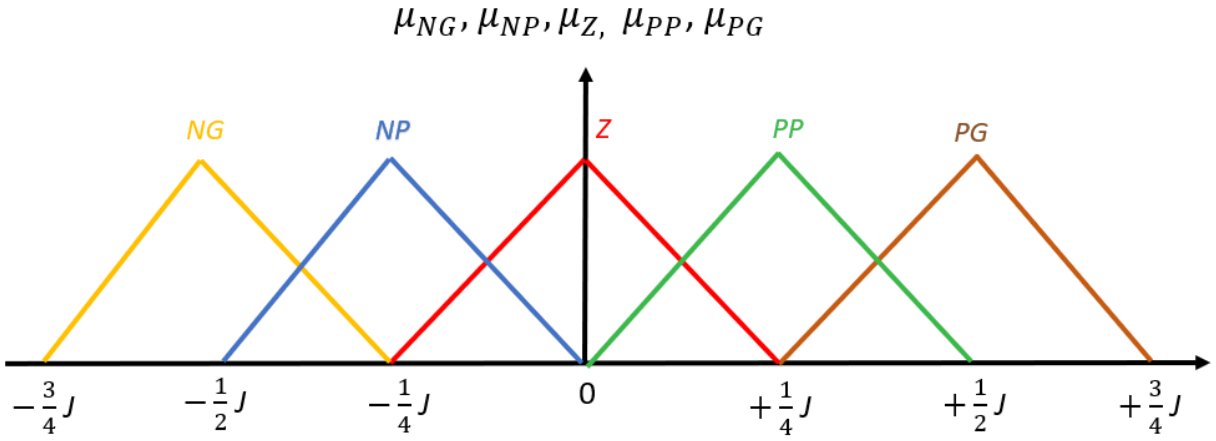


FIG. 4.14: Variável VJ - Partição *Fuzzy*.

Onde as funções NG, NP, Z, PP, PG significam Negativo Grande, Negativo Pequeno, Próximo à Zero, Positivo Pequeno e Positivo Grande, respectivamente.

O número máximo de regras que governam o sistema é dado pelo número de partições (k) de cada antecedente (n) (ROSS, 2009). Logo, pode-se obter até $k^n = 3^2 = 9$ regras. Nesse trabalho, obteve-se 5 regras reais. Os antecedentes Max_RTT e Avg_Est_RTT, serão substituídos por A e B, respectivamente, nas regras de produção. O consequente será a saída VJ resultante da aplicação das regras de produção listadas a seguir:

1. Se A e B são pequenos, então efetuar grande aumento de janela.
2. Se A é pequeno e B é médio, ou se A é médio e B é pequeno, então efetuar pequeno aumento de janela.
3. Se A e B são grandes, então efetuar grande redução de janela.
4. Se A é grande e B é médio ou pequeno, ou ainda se A é médio ou pequeno e B é grande, então efetuar pequena redução de janela.
5. Se A e B são médios, então efetuar pequeno aumento de janela.

As partições do sistema *fuzzy*, baseadas em regras de produção, podem ser resumidas na FAM (*Fuzzy Associative Memory*), conforme Tabela 4.3 .

As regras apresentadas fundamentam-se na experiência humana e fazem parte da construção do sistema *fuzzy*. Segundo Ross (2009), a percepção humana pode se basear na experiência, *expertise*, observação empírica, intuição ou conhecimento sobre o funcionamento de um sistema.

TAB. 4.3: Tabela FAM das Regras de Produção *fuzzy*

A/B	**P**	**M**	**G**
P	PG	PP	NP
M	PP	PP	NP
G	NP	NP	NG

O próximo passo é a realização da leitura *fuzzy* pelo método de inferência gráfica para verificar as entradas atuais $x_{1=\{R_1,R_2,\ R_3\}}$, onde cada entrada x poderá ativar a regra ou não, conforme Figura 4.15.

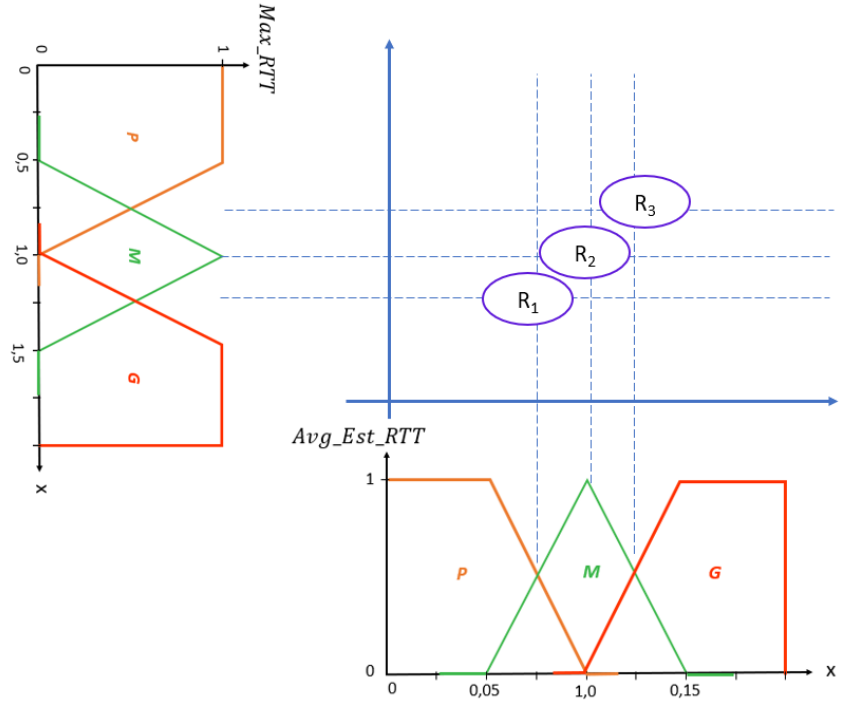


FIG. 4.15: Representação das Entradas Atuais com base em Ross (2009).

A ativação da regra dar-se-á mediante uma operação. A operação básica para processamento das regras de pertinência dos conjuntos *fuzzy*, segundo o modelo de Mamdani (ZADEH, 1976) para o processamento do conhecimento, ou seja, processamento dos antecedentes, operadores utilizados e consequentes, é dada pela Equação 4.16 a seguir:

$$\mu_{A\cap B} = min\left[\mu_A\left(x_i\right), \mu_B\left(x_i\right)\right] \tag{4.16}$$

Procede-se então à verificação das regras ativadas. Após disparo da regra, os diagnósticos dos consequentes da regra serão processados em conjunto para gerar uma resposta determinística para cada variável de saída do sistema (REZENDE, 2003). Por uma regra

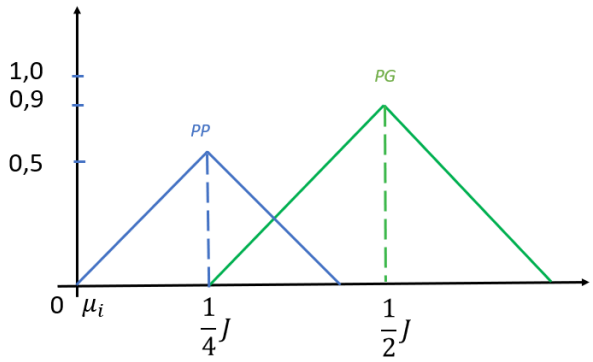


FIG. 4.16: *Método da Média Ponderada*

disparada, entende-se uma regra cujo processamento do antecedente, para as entradas atuais, gerou graus de pertinência não-nulos, ou seja, a relação *fuzzy* entre as entradas e os termos primários do antecendente é maior que zero. Com isso, completa-se a etapa de fuzzificação.

Por exemplo, se a variável A é P e a variável B é P, e o resultado da inferência é PG (Positivo Grande), então, de acordo com as regras de inferência, será realizado um grande aumento, ou seja, positivo grande, para o valor ou escalar obtido na etapa de defuzzificação, denominado VJ (Variação na Janela).

Na etapa de defuzzificação, deseja-se transformar a saída do processo *fuzzy* em uma quantidade escalar. Para a transformação em escalar, aplicou-se o método de defuzzificação denominado Método da Média Ponderada (ROSS, 2009). Esse método deve somente ser utilizado caso as saídas das funções de pertinência sejam simétricas (ROSS, 2009). Sendo assim, obteve-se os resultados de defuzzificação, conforme Equação 4.17 .

$$z^* = \frac{\sum \mu_{VJ}(\bar{z}).\bar{z}}{\sum \mu_{VJ}(\bar{z})} \tag{4.17}$$

Onde, o somatório é dado pela soma dos centroides de cada função de pertinência $\mu_{VJ}(\bar{z})$, ponderados pelo valor máximo de relacionamento na função $(\bar{Z})$, como exemplo da Figura 4.16 e respectiva Equação 4.1.2.

$$z^* = \frac{\sum \mu_{VJ}(\bar{z}).\bar{z}}{\sum \mu_{VJ}(\bar{z})} = \frac{\frac{1}{4}J\,(0.5) + \frac{1}{2}J\,(0.9)}{0,5 + 0,9}$$

Onde z^* é o valor defuzzificado para a variação da janela de tempo.

Assim, em função do processamento das características do tráfego de rede, segundo regras de relacionamento bem definidas, ocorre o re-dimensionamento das janelas de forma dinâmica. Essa foi a estrutura básica do sistema com a lógica *fuzzy* empregado nesse

trabalho:

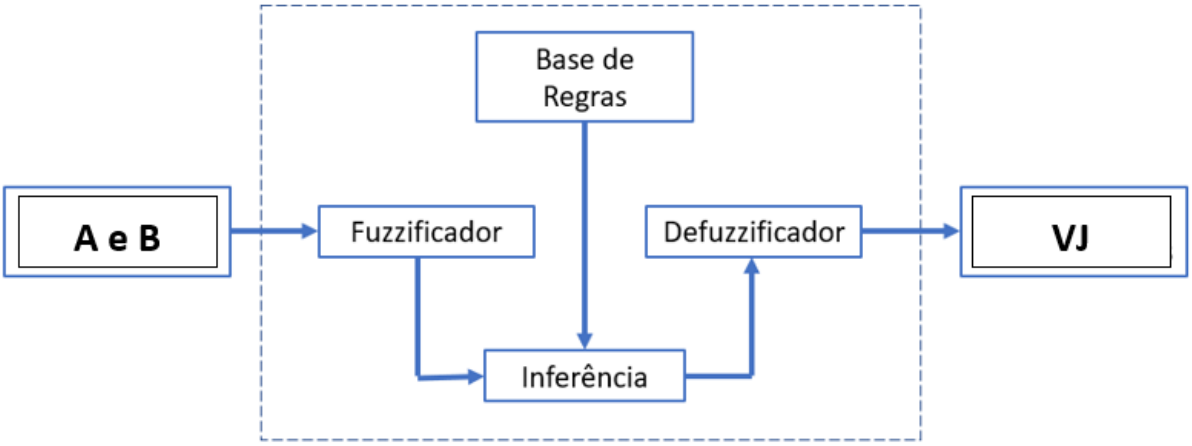


FIG. 4.17: Sistema *Fuzzy* pelo Método Mamdani com base em Zadeh (1976).

Onde A e B foram escolhidos, porque o ataque aproveita-se da leniência por parte de alguns servidores *Web* com relação aos atrasos na rede. Por este motivo, a lógica *fuzzy* foi utilizada. A detecção deve atentar para as mudanças de comportamento na rede e ajustar sua capacidade de análise para produzir resultados melhores.

4.2 O PREVISOR MÉDIA MÓVEL

Na análise exploratória de dados, fase inicial de estudo dos dados coletados em uma amostra, costuma-se utilizar técnicas que resumem e classificam o conjunto coletado para que se obtenham as informações pertinentes que serão utilizadas na fase final do processo, ou inferência estatística (LOPES, 2001).

A determinação do tamanho da amostra adequado é uma tarefa difícil, porque nunca o tamanho de uma amostra será igual ao tamanho original. No entanto, a decisão sobre qual seria o melhor tamanho, deverá ser a seguinte: o maior que se possa conseguir, mas deve-se levar em conta possíveis custos de processamento e ou possíveis riscos da amostra não ser representativa o suficiente para fornecer boas previsões. A Figura 4.18 ilustra que a decisão, quanto a escolha do tamanho da amostra, está associada a possíveis riscos e custos.

Compreende-se que, quanto maior for o tamanho da amostra, mais esta se aproxima da população e menor o erro que se pode cometer ao se fazer uma inferência (LOPES, 2001).

Mas, nem sempre é possível coletar a quantidade de dados que se deseja para a amostra. O importante é que essa amostra seja representativa da população.

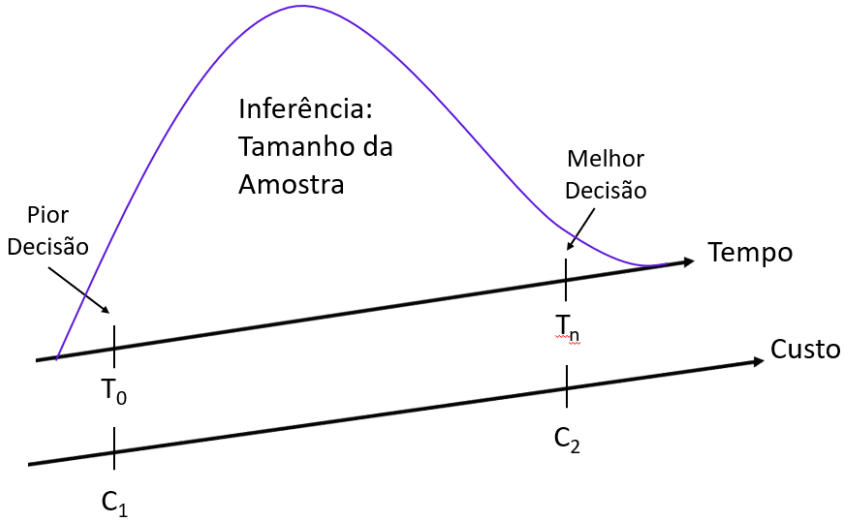


FIG. 4.18: Gráfico comparativo para tomada de decisão.

As medições servirão para alertar sobre anomalias relativas aos ataques *Slow DoS*. Portanto, na metodologia proposta não existem limiares previamente arbitrados, como por exemplo, limitações do número de conexões por IP. No entanto, essa metodologia apresenta a desvantagem de que, os valores iniciais por não possuírem histórico suficiente para processamento da métrica, não traz em um primeiro momento, valores assertivos.

O método de amostragem consistiu na escolha de um subconjunto com tráfego sem ataque, para não contaminar os resultados iniciais. Tal como ocorre com os algoritmos de aprendizado de máquina, existe um conjunto de exemplos de treinamento e um conjunto de teste. Para a avaliação de um preditor, estes conjuntos deverão ser distintos, para que não sejam produzidos resultados superestimados.

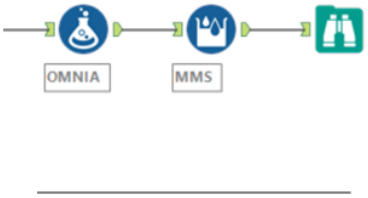
Para a previsão da série temporal a curto prazo, com base nas métricas F.O.R., optou-se pela média móvel simples como previsor da metodologia. A série de observações mais recentes, para cada uma das métricas, em um dado instante, fornecerá essas previsões. A previsão P está representada na Equação 4.18 para cálculo da Média Móvel Simples (MMS).

$$P_{j+1} = \frac{x_j + x_{j-1} + \ldots + x_{j-n+1}}{n} \tag{4.18}$$

Nesse trabalho, experimentou-se dois tipos períodos para a média móvel: o pré-determinado e o variável.

4.2.1 PERÍODO PRÉ-DETERMINADO PARA MMS

Para o cálculo da MMS, as 30 observações mais recentes foram utilizadas. O trecho do *workflow* científico, na Figura 4.19, mostra o cálculo da métrica *Omnia*, fornecendo dados para o cálculo da MMS, onde os respectivos resultados da MMS com base na métrica Omnia estão representados em tabela.



Janela_group30seg	Somatorio	Omnia	MMS
235	0	0	0
236	0	0	0
237	578	19	0
238	98	3	0.633333
239	93	3	0.733333
240	83	3	0.833333
241	0	0	0.933333
242	0	0	0.933333
243	0	0	0.933333
244	0	0	0.933333

FIG. 4.19: Processamento da Métrica Omnia com a MMS com 30 observações.

4.2.2 PERÍODO VARIÁVEL PARA MMS POR ERRO QUADRÁTICO MÉDIO

Conforme já mencionado, o tamanho amostral n deverá ser representativo da população N.

Nesse trabalho não se utiliza uma amostra probabilística e sim uma amostra intencional. Nesse tipo de amostragem há uma escolha deliberada dos elementos da população para constituírem a amostra. Entretanto, mesmo sendo deliberada, procurou-se observar o erro associado à escolha do intervalo para cálculo da MMS.

Deve-se tentar selecionar um valor para n que minimize o erro de previsão (MENASCÉ; ALMEIDA, 2003), que é definido pelo quadrado da diferença entre os valores real (r_j) e previsto (P_j), de acordo com a Equação 4.19 .

$$EQM = \frac{\sum_{j=1}^{n} (r_j - P_j)^2}{n} \tag{4.19}$$

4.3 A DESIGUALDADE TCHEBYCHEFF NA ANÁLISE DE DETECÇÃO

Segundo (SAVAGE, 1961), as desigualdades de Tchebycheff são úteis para o trabalho com distribuições, cuja forma da função de distribuição é desconhecida. Por esse motivo,

dado que não se conhece a distribuição das variáveis de tráfego envolvidas nesse trabalho, tais como aquelas calculadas pela métrica Flags e pela métrica RTTR, emprega-se a desigualdade Tchebycheff. Conforme verificação realizada no Apêndice 1 desse trabalho, essas variáveis não apresentam distribuição de dados próxima à uma distribuição normal.

A desigualdade Tchebycheff provê limites para a probabilidade de certos eventos, ou seja, ela fornece estimativas para os desvios da média em termos destes momentos. Tais momentos são calculados em função de uma variável aleatória e a natureza da variável aleatória deve ser especificada, isto é, deve-se dizer se a variável aleatória é uma soma, uma média ou alguma outra função.

Tudo o que é necessário para o uso dessas desigualdades são boas estimativas de certos momentos da população (SAVAGE, 1961).

A desigualdade Tchebycheff para uma variável está descrita pela Equação 4.20 .

$$P(X \geq \lambda\sigma + \mu \quad ou \quad X \leq \mu - \lambda\sigma) \leq \frac{1}{\lambda^2} \tag{4.20}$$

Onde, a variável aleatória X representa o valor estimado pelo previsor MMS. Admite-se que o valor esperado da média amostral $\overline{x}$ é igual a μ e o desvio padrão amostral será estimado pela correção para populações finitas, conforme Equação 4.21.

$$S = \frac{\sigma \cdot n}{(n-1)} \tag{4.21}$$

Dessa forma, pode-se então, utilizar a desigualdade Tchebycheff. Essa desigualdade define que a probabilidade dos grandes desvios em relação à média é muito pequena, significando que decresce segundo a razão $\frac{1}{\lambda^2}$. No entanto, esse é um estimador pobre para a probabilidade de grandes desvios em uma distribuição normal e para grandes valores de λ. Logo para valores intermediários é uma boa aproximação dos limites da distribuição (SAVAGE, 1961).

Se a estimativa de σ é razoavelmente boa, então essa desigualdade pode ser usada para testar hipóteses sobre a média e para encontrar intervalos de confiança para a média. Em muitas aplicações industriais, essa desigualdade é usada para estimar quanto da produção estará próximo da média do processo, onde se tem uma boa ideia da variância.

Pela desigualdade Tchebycheff, estima-se que pelo menos 89% do tráfego normal ficará dentro dos limiares de $\pm 3s$ a partir da média $\bar{X}$. Há uma chance de 89% do intervalo conter a média da população e consequentemente o que estiver fora dos limites será considerado anormal. Adotou-se portanto uma posição mais conservadora com o valor de $\lambda = 3$,

com base no controle estatístico de qualidade, cuja convenção dos limites de controle dos pontos do processo posiciona-se em 3σ a partir da média.

4.4 O ALGORITMO *T-SLOW*

O processamento do algoritmo principal *T-Slow* (Tchebycheff-Slow) computará o resultado da detecção a partir da entrada de outros algoritmos, representados aqui pelas funções F.O.R.

Apresenta-se a estrutura dos algoritmo *T-Slow*, com suas principais funções, para melhor entendimento da arquitetura para detecção apresentada na Figura 4.1.

Algorithm 1 T-*Slow*

1: $Pacotes[] \leftarrow Trafego[][]$
2: **if** $Flags(Pacotes[])= 1$ OR $Omnia\ (Pacotes[])= 1$ OR $RTTR\ (Pacotes[])= 1$ **then**
3: **return** Tráfego anormal.
4: **end if**
5: **return** 0

Conforme representação do Algoritmo 1, observa-se que as funções do *T-Slow* são executadas em paralelo. Os *pacotes* pertencentes a uma determinada janela de tempo serão submetidos às funções F.O.R. e a desigualdade Tchebycheff será responsável pela classificação da janela de tempo em anormal, retornando o valor um; e zero, caso contrário.

A implementação dos algoritmos propostos nesse trabalho foi realizada com o *software* para análise de *big data* chamado *Alteryx Designer*.

4.4.1 A FUNÇÃO *FLAGS*

Essa função retorna as janelas cuja anomalia foi detectada segundo as características da métrica *Flags*, conforme Algoritmo 2.

Algorithm 2 Função *Flags*

```
1: Pacotes[] ← Trafego[][]
2: function MEDIA(v)
3:     Calcula a média móvel da variável v para n períodos
4: end function
5: function DESVIO(v)
6:     Calcula o desvio padrão da variável v para n períodos
7: end function
8: function FLAGS
9:     for all Pacotes[] ∈ Trafego[][] do
10:        if "SynAck" ∈ Pacotes[] then
11:            F_SynAck[n] ← 1
12:        else if FinAck ∈ Pacotes[] then
13:            F_FinAck[n] ← 1
14:        end if
15:        Flags[n] ← F_SynAck[n] − F_FinAck[n]
16:    end for
17:    for all Janelas[j][] ∈ Trafego[][] do
18:        for all Pacotes[i][] ∈ Janela[j][] do
19:            Sum_Flags[j] ← Sum_Flags[j] + Flags[i]
20:            Sum_SynAck[j] ← Sum_SynAck[j] + FsynAck[i]
21:        end for
22:    end for
23:    for all Janela[j][] ∈ Trafego[][] do
24:        Media_Flags[j] ← Media(Sum_Flags[j])
25:        Desvios_Flags ← Desvio(Sum_Flags[j])
26:        Media_SynAck ← Media(Sum_SynAck[j])
27:        Desvios_SynAck ← Desvio(Sum_SynAck[j])
28:        if Sum_SynAck[j] ≥ Media_SynAck[j] + 3 ∗ Desvios_SynAck[j] AND
   Sum_Flags[j] ≥ Media_Flags[j] + 3 ∗ Desvio_Flags[j] then
29:            return 1
30:        end if
31:    end for
32:    return 0
33: end function
```

Para todo pacote de tráfego, o algoritmo analisa se há *flags* SYNACK ou FINACK no respectivo pacote e caso positivo, salva o valor 1 (um) nos vetores $F_SynAck[n]$ e $F_FinAck[n]$, respectivamente, onde n representa o número do pacote analisado.

O algoritmo cria um terceiro vetor $Flags[n]$, que receberá a diferença dos vetores F_SynAck e F_FinAck. Para cada janela, é calculada a soma do vetor $Flags[i]$ e $F_SynAck[i]$, onde i representa o pacote da janela analisada, pois cada janela consistirá de vários pacotes. Para cada janela, será calculada a média móvel das janelas mais recentes e o respectivo desvio padrão, tanto para a soma do vetor *Flags* ($Sum_Flags[j]$), como

para a soma do vetor *SynAck* ($Sum_SynAck[j]$).

Aplica-se a desigualdade Tchebycheff a estas duas variáveis. Então, a função *Flags* retornará 1 (um), caso a anormalidade seja detectada na janela, por ambas variáveis; e 0 (zero), caso contrário.

4.4.2 A FUNÇÃO *OMNIA*

Os pacotes incompletos deverão ser identificados e comparados às regras estabelecidas pelo teorema Tchebycheff, o Algoritmo 3 deverá alertar caso a anormalidade seja detectada.

Algorithm 3 Função *Omnia*

```
1: Pacotes[] ← Trafego[][]
2: function MEDIA(v)
3:     Calcula a média móvel da variável v para n períodos
4: end function
5: function DESVIO(v)
6:     Calcula o desvio padrão da variável v para n períodos
7: end function
8: function OMNIA
9:     for all Janelas[j][] ∈ Trafego[][] do
10:        for all Pacotes[i][] ∈ Janela[j][] do
11:            Omnia[j] ← Omnia[j] + Final_Incompleto[i] ▷ Somatório de pacotes com final incompleto dentro da janela j
12:        end for
13:    end for
14:    for all Janela[j][] ∈ Trafego[][] do
15:        Media_Omnia[j] ← Media(Omnia[j])
16:        Desvio_Omnia[j] ← Desvio(Omnia[j])
17:        if Omnia[j] ≥ Media_Omnia[j] + 3 * Desvio_Omnia[j] then
18:            return 1
19:        end if
20:    end for
21:    return 0
22: end function
```

O vetor $Omnia[j]$ contabiliza a quantidade de pacotes incompletos dentro de uma janela *j*. É calculada a média móvel do somatório de pacotes incompletos das janelas mais recentes e o respectivo desvio padrão. Então, a desigualdade Tchebycheff será aplicada e a função *Omnia* retorna 1 (um) caso a anormalidade seja detectada e 0 (zero), caso contrário.

4.4.3 A FUNÇÃO *RTTR*

Essa função aponta os valores de RTT Real (RTT) anormais para a tarefa de detecção do método proposto. Os RTT anormais compõem um somatório e a média dessa variável será confrontada com a média móvel dos últimos RTT por janela de acordo com a regra Tchebycheff.

Algorithm 4 Função *RTTR*

```
1: Pacotes[] ← Trafego[][]
2: function MEDIA(v)
3:     Calcula a média móvel da variável v para n períodos
4: end function
5: function DESVIO(v)
6:     Calcula o desvio padrão da variável v para n períodos
7: end function
8: function RTTR
9:     for all Janelas[j][] ∈ Trafego[][] do
10:        for all Pacotes[i][] ∈ Janela[j][] do
11:            Sum_RTT[j] ← Sum_RTT[j] + RTT[i]
12:        end for
13:        Media_RTT_Janela[j] ← Sum_RTT[j] ÷ n[j]      ▷ Calcula média de RTT dos pacotes da janela j
14:    end for
15:    for all Janela[j][] ∈ Trafego[][] do
16:        Media_RTT[j] ← Media(Media_RTT_Janela[j])
17:        Desvio_RTT[j] ← Desvio(Media_RTT_Janela[j])
18:        if Media_RTT_Janela[j] ≥ Media_RTT[j] + 3 * Desvio_RTT[j] then
19:            return 1
20:        end if
21:    end for
22:    return 0
23: end function
```

Para todos os pacotes de tráfego pertencentes a uma determinada janela de tempo, o vetor $Sum_RTT[j]$ produz uma somatório dos valores de RTT para cada janela de tempo. O vetor $Media_RTT[j]$ contém o somatório de RTT ($Sum_RTT[j]$) dividido pela quantidade n de pacotes existentes na janela. A detecção será realizada pela desigualdade Tchebycheff e a função *RTTR* retorna 1 (um) caso a anormalidade seja detectada e 0 (zero), caso contrário.

5 ESTUDO DE CASO: *CIC DOS DATASET*

CIC são as iniciais para *Canadian Institute for Cybersecurity*. Esta instituição, localizada na universidade de New Brunswick no Canadá, forneceu o *dataset* para a realização do presente estudo de Caso.

O estudo de caso foi o método de pesquisa escolhido para avaliar as medidas de desempenho do algoritmo *T-Slow* com a metodologia proposta e comparar esses resultados com outras abordagens.

Conforme já mencionado, a acurácia de detecção proposta nesse trabalho apoia-se no poder das métricas, na estimação das janelas de tempo mediante a lógica *fuzzy* e também no intervalo de confiança estipulado pela desigualdade Tchebycheff. Com a aplicação dos algoritmos foi possível observar a aptidão do algoritmo proposto na detecção dos ataques *Slow DoS*.

A seleção das métricas corretas é uma tarefa difícil, mas não somente as métricas impactam os resultados. Outras questões podem influenciar o desempenho de um algoritmo de detecção, tais como, o local onde o mecanismo de defesa deverá atuar, o dinamismo do sistema (eficiência), a adaptabilidade a novos ataques e a qualidade do *dataset*.

Para a pesquisa, o mais importante dos fatores listados é a qualidade do *dataset*.

5.1 O *DATASET*

Do ponto de vista da detecção de ataques, dois parâmetros foram levados em consideração para a escolha do *dataset*: As implementações utilizadas no ataque e a qualidade do mesmo. Sobre a qualidade do *dataset*, é importante que este não possua viés, isto é, o tráfego de rede deve ser capturado corretamente (BHUYAN et al., 2014).

Há grande dificuldade na obtenção de *dataset* privado para a pesquisa de detecção de ataques, pois há grande receio de danificar ativos importantes durante a aplicação do experimento.

Sobre o tema *Slow DoS*, existem poucos *datasets* que possam ser explorados. O *CIC DoS Dataset (Application-Layer)* foi escolhido porque o *Canadian Institute for Cybersecurity* (CIC) possui alguns dos *datasets benchmarks* mais utilizados em detecção de anomalias, sendo utilizados em todo o mundo por universidades, indústria privada e pesquisadores independentes.

Segundo o CIC-DoS (2017), as características *dataset* são as seguintes:

- Apache Linux v.2.2.22;
- Traces do ISCX-IDS Dataset (livre de ataques);
- Produção de 4 tipos de ataques à camada de aplicação com diferentes ferramentas;
- 8 traces de ataque DoS à camada de aplicação com 24 horas de tráfego de rede totalizando 4.6 GB; e
- Ataques direcionados aos 10 Servidores Web do ISCX-IDS Dataset.

As variações de ataques *Slow DoS* presentes no *dataset* são: *Slow Headers*, *Slowloris*, *Slow Body*, *R.U.D.Y.*, *Slow Read*, conforme Figura 5.1. Constam também outros ataques de inundação tradicional, como o *H.U.L.K.*, *Golden Eye* e DDossim (JAZI et al., 2017).

TAB. 5.1: Lista de Ataques com base no CIC-DoS (2017).

Ataque	Alvo	Início
Slow Body	75.127.97.72	após 00:53 minutos
Slow Read	75.127.97.72	após 01:58 minutos
Slow Headers	74.63.40.21	após 02:57 (177 minutos)
R.U.D.Y.	75.127.97.72	após 03:08 (188 minutos)
R.U.D.Y.	208.113.162.153	após 03:29 (209 minutos)
Slow Headers	67.220.214.50	após 06:00 (360 minutos)
Slow Body	69.192.24.88	após 08:13 (493 minutos)
Slow Body	97.74.144.108	após 09:03 (543 minutos)
Slow Body	203.73.24.75	após 09:09 (549 minutos)
R.U.D.Y.	97.74.144.108	após 09:20 (560 minutos)
Slow Read	74.55.1.4	após 11:02 (662 minutos)
Slow Headers	97.74.104.201	após 11:27 (687 minutos)
Slowloris	97.74.144.108	após 15:20 (920 minutos)
Slow Headers	97.74.144.108	após 15:47 (947 minutos)
Slowloris	75.127.97.72	após 16:33 (993 minutos)
Slow Headers	75.127.97.72	após 17:13 (1033 minutos)
R.U.D.Y.	74.55.1.4	após 20:59 (1259 minutos)

O CIC DoS Dataset possui 4 tipos de ataques: Golden Eye, DDoSsim, H.U.L.K e Slow Rate. Os ataques do tipo *Slow Rate* ou *Slow DoS* foram implementados pelas ferramentas SlowHTTPTest, Slowloris, R.U.D.Y., conforme ilutstrado na Figura 5.1. Cada uma das implementações, eventualmente empresta seu nome ao ataque por ela realizado.

Os ataques *Golden Eye*, DDoSsim e H.U.L.K não são abordados nesse trabalho.

O *dataset* possui 24 horas de tráfego e os ataques foram direcionados à 10 servidores. E, assim que o servidor tornou-se sem resposta, os ataques foram interrompidos.

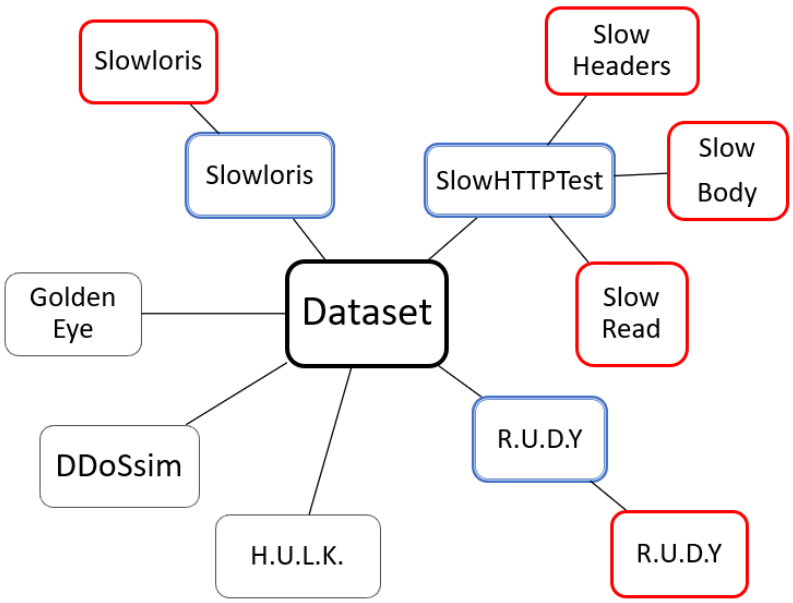


FIG. 5.1: Implementações e nomes atribuídos aos ataques do tipo *Slow Dos* presentes no CIC-DoS (2017) *Dataset*.

5.2 RESULTADOS OBTIDOS

A principal particularidade dos ataques lentos à camada de aplicação é a furtividade. Por exemplo, o ataque *Slowloris* caracteriza-se pelo envio de tráfego malicioso em intervalos regulares (AQIL et al., 2015). É justamente essa peculiaridade que incrementa o poder do ataque, fazendo-o muito difícil de ser detectado, se comparado a um ataque tradicional por inundação.

Esse trabalho destaca o método de dimensionamento das janelas, as métricas e a análise da desigualdade Tchebycheff com a expectativa de resultados melhores na detecção de ataques *Slow DoS*. São raras as discussões mais detalhadas sobre quais janelas de tempo seriam as mais adequadas para a análise de tráfego de rede por se tratar de um sistema complexo e variante ao longo do tempo (DOS SANTOS, 2009).

A seguir, apresenta-se os resultados das funções F.O.R. do algoritmo *T-Slow*, segundo 2 estratégias distintas:

1) Janelas Estáticas: Onde o tamanho das janelas de tempo são diretamente proporcionais ao tráfego, e portanto, com intervalos iguais; e

2) Janelas *Fuzzy*: Onde o tamanho das janelas de tempo são determinadas pela lógica *fuzzy*, e portanto, apresentam intervalos variáveis.

Considerando a escala de tempo estimada em cada uma das abordagens a identificação

dos pacotes anormais dar-se-á em um instante de tempo t_j. Define-se por j a j-ésima janela de tempo, comparada ao intervalo imediatamente inferior, t_{j-1}. A janela de tempo é classificada em dois casos: normal, caso o valor da métrica no instante t do intervalo considerado esteja livre de ataque ou anormal, caso contrário.

O tráfego é marcado com anormal, caso a anormalidade seja detectada por alguma das funções do algoritmo *T-Slow: Flags, RTTR* ou *Omnia*.

Na estratégia 1, a funções trabalham com janelas de tempo estáticas, conforme Figura 5.2, onde cada janela de tempo equivale a 30 segundos. Obteve-se 2881 janelas de tempo para o tráfego do CIC-DoS (2017) *Dataset*.

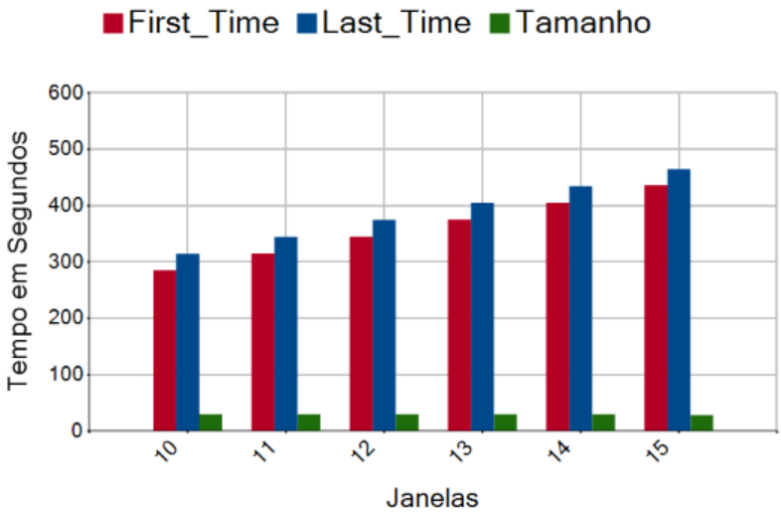


FIG. 5.2: Distribuição Uniforme para o Tamanho das Janelas de Tempo.

Na estratégia 2, essas funções orientam-se pela variação das janelas determinadas pela lógica *fuzzy*. Mediante a dinâmica *fuzzy*, obteve-se 1277 janelas para o tráfego do CIC DoS *Dataset*. Apresenta-se abaixo a dinâmica *fuzzy* na variação do tamanho das janelas de tempo (em segundos), durante o período compreendido entre dois ataques (Slow Body e Slow Read) presentes no dataset AppDDoS, conforme Figura 5.3.

O ataque *Slow Body* teve início após 53 minutos do início da captura do tráfego e a janela correspondente ao tempo é a de número 29. O próximo ataque é o *Slow Read* que inicia-se 1 hora e 58 minutos após o início da captura do tráfego e janela correspondente é a de número 58. Observa-se que o tamanho da janela após o início do ataque diminui significativamente após um determinado tempo, tanto para a janela 29, quanto para a janela 58, conforme 5.3. Esta Figura representa a forma como a lógica *fuzzy* realiza a diminuição das janelas de tempo ajustando-as, de forma a detectar as anormalidades com maior acurácia.

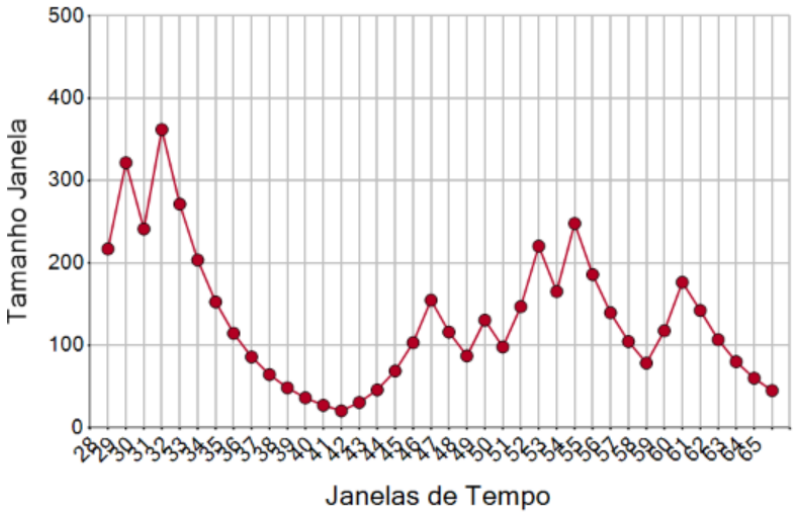


FIG. 5.3: Variação das Janelas de Tempo *Fuzzy*.

5.2.1 RESULTADOS COM A MÉTRICA *FLAGS*

Observa-se no gráfigo da Figura 5.4 que dentre as três séries, as mudanças expressivas estão nos valores do somatório da *flag* SYN ACK. Outro fato a destacar é que um súbito aumento no pedido de conexões causou altos valores para a diferenças entre as *flags* SYN ACK e FIN ACK.

Com a métrica *Flags*, segundo a estratégia 1, foram detectadas anomalias com as características análogas aos ataques: *Slow Body*, *Slow Read* e *Slow Headers*. E, com a estratégia 2, além desses ataques, também foi detectado o *Slowloris*. A Figura 5.4 1 aponta a anormalidade na janela 108, correspondente ao ataque *Slow Body* de acordo com a Tabela 5.1.

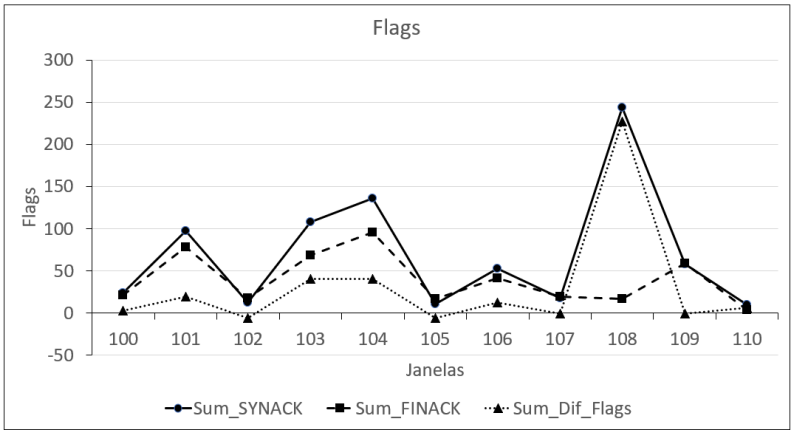


FIG. 5.4: Gráfico para a distribuição dos valores máximos para *Flags* para janelas de tempo estáticas.

5.2.2 RESULTADOS COM A MÉTRICA OMNIA

Essa métrica, destinada à detecção dos pacotes incompletos, apontou duas janelas portadoras dessa anomalia, conforme Figura 3.3. As anomalias foram detectadas para as janelas 237 e 1325 de acordo com lista de ataques da Tabela (5.1). A Figura 5.5 mostra a anomalia na janela 237.

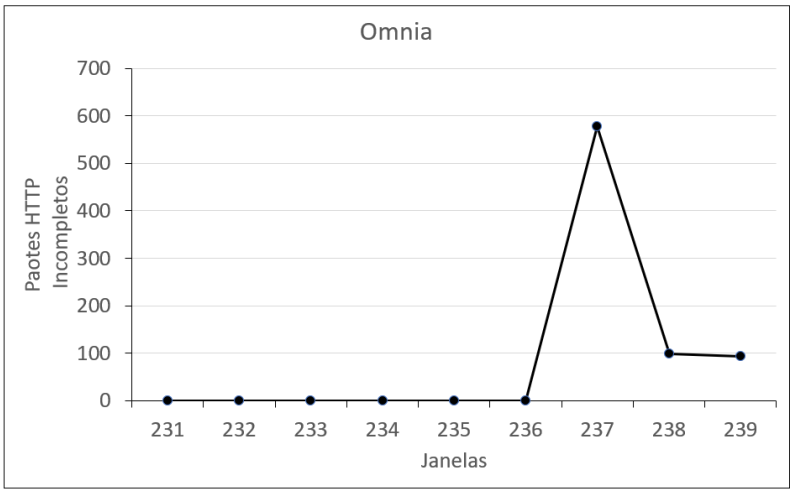


FIG. 5.5: Distribuição dos somatórios de requisições incompletas (sem CRLF CRLF) por janela de tempo estática.

O ataque *Slow Read* foi detectado, porém as características desse ataque presentes na literatura até o momento, não se enquadram no projeto da métrica *Omnia*, que foi desenhada para identificar requisições incompletas por janela de tempo. A característica fundamental do Slow Read é a resposta lenta da vítima às solicitações GET do cliente. Tal característica se deve ao fato de que as requisições do cliente apresentam cabeçalho com o método GET completo e a estratégia de ataque é que, inicialmente, o tamanho da janela de recepção é bem alto durante o *three-way handshake*, mas quando a conexão é estabelecida o tamanho da janela de recepção cai drasticamente, o que obriga ao servidor enviar *payloads* de tamanho mínimo a um ritmo muito lento, causando ao servidor reservar recursos preciosos com requisições ilegítimas que podem leva-lo à negação do serviço aos usuários legítimos.

Com a estratégia 1, a métrica detectou anomalias pertencentes apenas ao ataque *Slow Read*. Com a estratégia 2, além do *Slow Read*, anomalias pertencentes aos ataques *Slow Body* e *Slow Headers*.

5.2.3 RESULTADOS COM A MÉTRICA RTTR

Nesse trabalho, o somatório do RTT real foi comparado à média dos RTT da últimas janelas para a tarefa de detecção.

A Figura 5.6 demonstra esse tipo de anomalia já na janela de número 108.

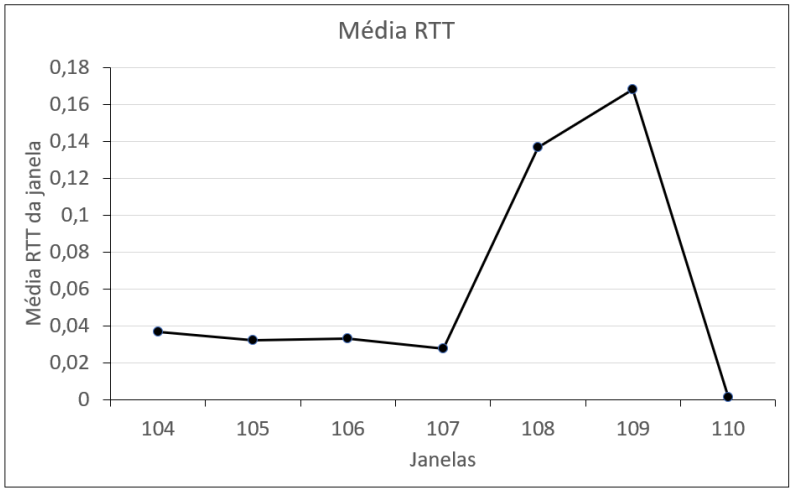


FIG. 5.6: Gráfico para a distribuição dos valores da média RTTR por janela de tempo estática.

Como os valores de RTT se concretizam na fonte, ou seja, o emissor do pacote, pode-se confirmar a ocorrência dos seguintes ataques com a estratégia 1 e 2, conforme Tabela 5.1:

- Slow Body;
- Slow Read;
- Slow Headers;
- Slowloris e;
- R.U.D.Y.

Em função da verificação do menor EQM para determinação do tamanho da amostra para o previsor, conforme Equação 4.19, os resultados obtidos foram otimizados, no que tange à ocorrência de verdadeiros positivos. A nova escolha do período da amostra para o cálculo do previsor em ambas as estratégias proporcionou melhor índice de detecção de ataques ao passo que também reduz o processamento do algoritmo, pois indica uma quantidade menor janelas amostradas. As Tabelas 5.2 e 5.3 mostram os novos dados para janelas amostradas com aplicação EQM na estratégias 1 e 2.

TAB. 5.2: Valor de n que minimizará o erro de previsão - Estratégia 1.

Métrica	Menor EQM	n
Flags	0,00619	10
RTTR	0,00029	7
Omnia	0	10

TAB. 5.3: Valor de n que minimizará o erro de previsão - Estratégia 2.

Métrica	Menor EQM	n
Flags	0,000303	8
RTTR	0,000231	9
Omnia	5,093662	28

É importante destacar que o cálculo do valor ideal de n deve ser realizado em tráfego livre de ataque, para que não haja contaminação dos resultados do previsor. O método de previsão por MM adequa-se à metodologia proposta nesse trabalho, uma vez que a característica principal da MM é que apenas um valor previsto poderá ser calculado por vez, e por essa razão é indicada para previsões de curto prazo.

5.3 ANÁLISES DE DESEMPENHO

Segundo Gupta e Badve (2017) nenhum mecanismo de defesa pode alcançar 100% de detecção dos pacotes de ataque. Mas, deve alcançar altas taxas de verdadeiros positivos e verdadeiros negativos, com o mínimo possível de falsos positivos e falsos negativos.

Nesse trabalho, dentre as medidas de desempenho para algoritmos de detecção, considerou-se a taxa de erro na classe positiva como a medida de desempenho mais relevante.

É importante destacar que, caso a métrica se baseie apenas no padrão do tráfego de rede, qualquer *outlier* poderia ser detectado como ataque. Desse fato observa-se que, dependendo do ambiente de rede, que pode ser bastante instável, este pode contribuir para o aumento de falsos positivos. Apesar do planejamento das métricas preocupar-se com essa possibilidade, o fato é que os ambientes de rede são instáveis por natureza e a ocorrência nula de falsos positivos, em tese, não faz parte da realidade. Por exemplo, um sistema de defesa pode ter zero falsos positivos com uma determinada rede em particular mas, não necessariamente terá o mesmo resultado em outros ambientes de rede.

As métricas F.O.R. baseiam-se não só no comportamento da rede, mas também nas anomalias que os ataques *Slow DoS* proporcionam ao ambiente de rede.

A metodologia proposta nessa dissertação prevê o uso da lógica *fuzzy* para a estimação

dos intervalos de tempo, previsor MMS e avaliação de detecção segundo a desigualdade Tchebycheff.

Mas, a fim de verificar os resultados alcançado pela metodologia proposta, foi realizado um estudo comparativo da análise de desempenho desta com outras abordagens. O estudo comparativo propõe a análise de desempenho com 4 abordagens distintas, onde a abordagem 4 representa a metodologia proposta. São elas:

1. Janelas estáticas com amostragem de 30 janelas;
2. Janelas estáticas com amostragem por EQM;
3. Janelas *fuzzy* com amostragem de 30 janelas e;
4. Janelas *fuzzy* com amostragem por EQM.

Para avaliação desses resultados, será utilizada a matriz confusão para problemas de duas classes, conforme 5.4.

TAB. 5.4: Avaliação do modelo preditivo para problema com duas classes com base em Faceli et al. (2015).

Matriz Confusão		Predita	
		+	-
Classe Verdadeira	+	VP	FN
	-	FP	VN

A matriz confusão traz informações entre a classe verdadeira e a classe predita. A classe pode ser positiva (+), quando há ataque, conforme Tabela 5.1 e negativa (-), caso contrário. VP (verdadeiro positivo) é a quantidade de verdadeiros positivos, ou seja, janelas detectadas corretamente com ataque; FP (falsos positivos) são as janelas que foram detectadas com ataque, mas na realidade não contém ataque; VN (verdadeiros negativos) é a quantidade de janelas que não são ataques e também não foram detectados quaisquer ataques e FN é a quantidade de janelas que deveriam ter sido detectadas com ataque, mas não foram detectadas. Logo, o somatório de VP, VN, FP e FN corresponde à n (número de janelas de tempo em cada abordagem).

Assim, a regra comum aplicada para classificação dos resultados do algoritmo *T-Slow* em verdadeiros positivos ou falsos positivos foi a seguinte:

1. Quando o *T-Slow* identificou anomalia em janela ou janelas que coincidiam ou pertenciam à janela indicada pelo gabarito da Tabela 5.1, a anomalia foi identificada como VP; e

2. Quando o *T-Slow* identificou anomalias nas janelas imediatamente posteriores à janela indicada conforme o gabarito da Tabela 5.1, foi realizada uma apuração do tráfego dentro da janela. Se identificou-se que o comportamento suspeito (ex. demandas com alto RTTR, envio de pacotes incompletos, aumento expressivo de conexões) era atribuído ao endereço IP listado como alvo, conforme gabarito, a anomalia foi classificada como VP e FP, caso contrário.

Com auxílio da matriz confusão, serão calculadas também outras medidas de desempenho (FACELI et al., 2015) para o sistema de detecção, que aqui denota-se por $\hat{p}$. São elas:

- TFN - Taxa de falsos negativos ou taxa de erro na classe positiva, pela Equação 5.1;
- TFP - Taxa de falsos positivos ou taxa de erro na classe negativa, pela Equação 5.2;
- Acurácia - Taxa de acertos no total, pela Equação 5.3;
- Precisão - Proporção de janelas com ataque, detectadas corretamente sobre todas aquelas que foram preditas como positivas pelo sistema de detecção, pela Equação 5.4;e
- Sensibilidade - Taxa de acerto na classe positiva ou taxa de verdadeiros positivos pela Equação 5.5.

$$err_{+}(\hat{p}) = \frac{FN}{VP + FN} \tag{5.1}$$

$$err_{-}(\hat{p}) = \frac{FP}{FP + VN} \tag{5.2}$$

$$ac(\hat{p}) = \frac{VP + VN}{n} \tag{5.3}$$

$$prec(\hat{p}) = \frac{VP}{VP + FP} \tag{5.4}$$

$$sens(\hat{p}) = \frac{VP}{VP + FN} \tag{5.5}$$

5.3.1 JANELAS ESTÁTICAS COM MM FIXA

Para o cálculo das medidas de desempenho da detecção *T-Slow* com médias móveis simples, cujo período empregado foi o das 30 janelas de tempo mais recentes, foram considerados os resultados da Tabela 5.5.

TAB. 5.5: Janelas identificadas na Abordagem 1.

Janelas Identificadas pelo Algoritmo *T-Slow*				
Ataque	Alvo	Início	Janela	*T-Slow*
Slow Body	75.127.97.72	após 00:53 min	106	108, 109
Slow Read	75.127.97.72	após 01:58 min	236	237, 238, 239, 240
Slow Headers	74.63.40.21	após 02:57 (177 min)	354	371
R.U.D.Y.	75.127.97.72	após 03:08 (188 min)	376	∅
R.U.D.Y.	208.113.162.153	após 03:29 (209 min)	418	419
Slow Headers	67.220.214.50	após 06:00 (360 min)	720	725, 737
Slow Body	69.192.24.88	após 08:13 (493 min)	986	988, 989
Slow Body	97.74.144.108	após 09:03 (543 min)	1086	1088, 1089, 1093
Slow Body	203.73.24.75	após 09:09 (549 min)	1098	1106
R.U.D.Y.	97.74.144.108	após 09:20 (560 min)	1120	∅
Slow Read	74.55.1.4	após 11:02 (662 min)	1324	1325, 1326, 1328
Slow Headers	97.74.104.201	após 11:27 (687 min)	1374	1379, 1380, 1384, 1385, 1391
Slowloris	97.74.144.108	após 15:20 (920 min)	1840	1841, 1842, 1843, 1844
Slow Headers	97.74.144.108	após 15:47 (947 min)	1894	∅
Slowloris	75.127.97.72	após 16:33 (993 min)	1986	1987, 1989, 1990
Slow Headers	75.127.97.72	após 17:13 (1033 min)	2066	2072, 2083
R.U.D.Y.	74.55.1.4	após 20:59 (1259 min)	2518	2518, 2519

A matriz confusão correspondente a essa abordagem está representada na Tabela 5.6.

TAB. 5.6: Matriz Confusão - Abordagem 1

Matriz Confusão		Classe Predita	
		+	-
Classe Verdadeira	+	VP = 36	FN = 15
	-	FP = 128	VN = 2702

As medidas de desempenho foram as seguintes:

$$err_{+}(\hat{p}) = \frac{FN}{VP + FN} = \frac{15}{36 + 15} = 29\% \tag{5.6}$$

$$err_{-}(\hat{p}) = \frac{FP}{FP + VN} = \frac{128}{128 + 2702} = 5\% \tag{5.7}$$

$$ac(\hat{p}) = \frac{VP + VN}{n} = \frac{36 + 2702}{2881} = 95\% \tag{5.8}$$

$$prec(\hat{p}) = \frac{VP}{VP + FP} = \frac{36}{36 + 128} = 22\% \tag{5.9}$$

$$sens(\hat{p}) = \frac{VP}{VP + FN} = \frac{36}{36 + 15} = 71\% \tag{5.10}$$

Com essa abordagem, o algoritmo *T-Slow* obteve uma taxa de erro na classe positiva alta, correspondente à 29%. Portanto, nesse caso, o algoritmo deixou de classificar ataques de fato, com erro de 29% sobre o total de ataques. A quantidade alta de FN afetou o cálculo da sensibilidade. E, a precisão foi prejudicada pelo alto índice de FP. A acurácia obteve ótimo resultado com 95% de acertos no total da classificação do trafego sob análise.

5.3.2 JANELAS ESTÁTICAS COM MM VARIÁVEL POR EQM

Para o cálculo das medidas de desempenho da detecção *T-Slow* com médias móveis simples, cuja escolha do período empregado foi determinado pelo menor erro quadrático médio, foram considerados os resultados da Tabela 5.7.

TAB. 5.7: Janelas identificadas na Abordagem 2.

Janelas Identificadas pelo Algoritmo *T-Slow*				
Ataque	Alvo	Início	Janela	*T-Slow*
Slow Body	75.127.97.72	após 00:53 min	106	108, 109
Slow Read	75.127.97.72	após 01:58 min	236	237, 238, 239, 240
Slow Headers	74.63.40.21	após 02:57 (177 min)	354	360, 371
R.U.D.Y.	75.127.97.72	após 03:08 (188 min)	376	∅
R.U.D.Y.	208.113.162.153	após 03:29 (209 min)	418	418
Slow Headers	67.220.214.50	após 06:00 (360 min)	720	725, 726, 737
Slow Body	69.192.24.88	após 08:13 (493 min)	986	988,989
Slow Body	97.74.144.108	após 09:03 (543 min)	1086	1088,1093
Slow Body	203.73.24.75	após 09:09 (549 min)	1098	1106
R.U.D.Y.	97.74.144.108	após 09:20 (560 min)	1120	1120, 1121
Slow Read	74.55.1.4	após 11:02 (662 min)	1324	1325,1326,1327,1328
Slow Headers	97.74.104.201	após 11:27 (687 min)	1374	1379, 1380, 1391
Slowloris	97.74.144.108	após 15:20 (920 min)	1840	1841, 1842, 1843, 1844
Slow Headers	97.74.144.108	após 15:47 (947 min)	1894	∅
Slowloris	75.127.97.72	após 16:33 (993 min)	1986	1987, 1990
Slow Headers	75.127.97.72	após 17:13 (1033 min)	2066	2071, 2072, 2083
R.U.D.Y.	74.55.1.4	após 20:59 (1259 min)	2518	2518, 2519

A matriz confusão correspondente a essa abordagem está representada na Tabela 5.8.

TAB. 5.8: Matriz Confusão - Abordagem 2

Matriz Confusão		Classe Predita	
		+	-
Classe Verdadeira	+	VP = 38	FN = 14
	-	FP = 331	VN = 2498

As medidas de desempenho foram as seguintes:

$$err_{+}(\hat{p}) = \frac{FN}{VP + FN} = \frac{14}{38 + 14} = 27\% \tag{5.11}$$

$$err_{-}(\hat{p}) = \frac{FP}{FP + VN} = \frac{331}{331 + 2498} = 12\% \tag{5.12}$$

$$ac(\hat{p}) = \frac{VP + VN}{n} = \frac{38 + 2498}{2881} = 88\% \tag{5.13}$$

$$prec(\hat{p}) = \frac{VP}{VP + FP} = \frac{38}{38 + 331} = 10\% \tag{5.14}$$

$$sens(\hat{p}) = \frac{VP}{VP + FN} = \frac{38}{38 + 14} = 73\% \tag{5.15}$$

Nesta abordagem, o algoritmo *T-Slow* também obteve uma taxa de erro na classe positiva alta, correspondente à 27%. O único índice que melhorou, com relação à abordagem anterior, foi a sensibilidade que aumentou dois pontos percentuais.

5.3.3 JANELAS DINÂMICAS *FUZZY* COM MM FIXA

Para o cálculo das medidas de desempenho da detecção *T-Slow* com médias móveis simples, cujo período empregado foi o das 30 janelas de tempo mais recentes e o tamanho das janelas foi definido pela lógica *fuzzy*, foram considerados os resultados da Tabela 5.9.

A matriz confusão correspondente a essa abordagem está representada na Tabela 5.10.

As medidas de desempenho foram as seguintes:

$$err_{+}(\hat{p}) = \frac{FN}{VP + FN} = \frac{10}{19 + 10} = 34\% \tag{5.16}$$

$$err_{-}(\hat{p}) = \frac{FP}{FP + VN} = \frac{80}{80 + 1168} = 6\% \tag{5.17}$$

TAB. 5.9: Janelas identificadas na Abordagem 3.

Janelas Identificadas pelo Algoritmo *T-Slow*				
Ataque	Alvo	Início	Janela	*T-Slow*
Slow Body	75.127.97.72	após 00:53 min	106	∅
Slow Read	75.127.97.72	após 01:58 min	236	58, 59
Slow Headers	74.63.40.21	após 02:57 (177 min)	354	93
R.U.D.Y.	75.127.97.72	após 03:08 (188 min)	376	∅
R.U.D.Y.	208.113.162.153	após 03:29 (209 min)	418	∅
Slow Headers	67.220.214.50	após 06:00 (360 min)	720	250, 251
Slow Body	69.192.24.88	após 08:13 (493 min)	986	326, 328
Slow Body	97.74.144.108	após 09:03 (543 min)	1086	∅
Slow Body	203.73.24.75	após 09:09 (549 min)	1098	356
R.U.D.Y.	97.74.144.108	após 09:20 (560 min)	1120	∅
Slow Read	74.55.1.4	após 11:02 (662 min)	1324	412
Slow Headers	97.74.104.201	após 11:27 (687 min)	1374	422
Slowloris	97.74.144.108	após 15:20 (920 min)	1840	512
Slow Headers	97.74.144.108	após 15:47 (947 min)	1894	528
Slowloris	75.127.97.72	após 16:33 (993 min)	1986	542, 543, 546
Slow Headers	75.127.97.72	após 17:13 (1033 min)	2066	682, 684, 685, 686
R.U.D.Y.	74.55.1.4	após 20:59 (1259 min)	2518	1111

TAB. 5.10: Matriz Confusão - Abordagem 3

Matriz Confusão		Classe Predita	
		+	-
Classe Verdadeira	+	VP = 19	FN = 10
	-	FP = 80	VN = 1169

$$ac(\hat{p}) = \frac{VP + VN}{n} = \frac{19 + 1168}{1277} = 93\% \tag{5.18}$$

$$prec(\hat{p}) = \frac{VP}{VP + FP} = \frac{19}{19 + 80} = 19\% \tag{5.19}$$

$$sens(\hat{p}) = \frac{VP}{VP + FN} = \frac{19}{19 + 10} = 66\% \tag{5.20}$$

A utilização de períodos fixos para a média móvel associada à lógica *fuzzy* para a estimação das janelas não foi eficiente para nenhuma das medidas de desempenho, principalmente para a sensibilidade e para a taxa de erro na classe positiva. O algoritmo *T-Slow* obteve uma taxa de erro na classe positiva alta, correspondente à 34%. A sensibilidade foi prejudicada pelo alto índice de FN.

5.3.4 JANELAS DINÂMICAS *FUZZY* COM MM VARIÁVEL POR EQM

E, finalmente o cálculo das medidas de desempenho da detecção *T-Slow* com médias móveis simples, cuja escolha do período empregado foi determinado pelo menor EQM, e o tamanho das janelas foi definido pela lógica *fuzzy*, conforme os resultados da Tabela 5.11.

TAB. 5.11: Janelas identificadas na Abordagem 4.

Janelas Identificadas pelo Algoritmo *T-Slow*				
Ataque	Alvo	Início	Janela	*T-Slow*
Slow Body	75.127.97.72	após 00:53 min	29	29, 30
Slow Read	75.127.97.72	após 01:58 min	58	58, 59
Slow Headers	74.63.40.21	após 02:57 (177 min)	88	93
R.U.D.Y.	75.127.97.72	após 03:08 (188 min)	94	∅
R.U.D.Y.	208.113.162.153	após 03:29 (209 min)	99	∅
Slow Headers	67.220.214.50	após 06:00 (360 min)	245	251
Slow Body	69.192.24.88	após 08:13 (493 min)	326	326, 327
Slow Body	97.74.144.108	após 09:03 (543 min)	351	351, 353
Slow Body	203.73.24.75	após 09:09 (549 min)	355	356
R.U.D.Y.	97.74.144.108	após 09:20 (560 min)	358	∅
Slow Read	74.55.1.4	após 11:02 (662 min)	412	412
Slow Headers	97.74.104.201	após 11:27 (687 min)	419	421, 422
Slowloris	97.74.144.108	após 15:20 (920 min)	512	512
Slow Headers	97.74.144.108	após 15:47 (947 min)	524	525
Slowloris	75.127.97.72	após 16:33 (993 min)	544	542, 544, 546
Slow Headers	75.127.97.72	após 17:13 (1033 min)	671	675, 682, 684, 685
R.U.D.Y.	74.55.1.4	após 20:59 (1259 min)	1111	1111

A matriz confusão correspondente a essa abordagem está representada na Tabela 5.12.

TAB. 5.12: Matriz Confusão - Abordagem 4

Matriz Confusão		Classe Predita	
		+	-
Classe Verdadeira	+	VP = 30	FN = 3
	-	FP = 105	VN = 1140

As medidas de desempenho foram as seguintes:

$$err_{+}(\hat{p}) = \frac{FN}{VP + FN} = \frac{3}{30 + 3} = 9\% \tag{5.21}$$

$$err_{-}(\hat{p}) = \frac{FP}{FP + VN} = \frac{105}{105 + 1140} = 8\% \tag{5.22}$$

$$ac(\hat{p}) = \frac{VP + VN}{n} = \frac{30 + 1139}{1277} = 92\% \tag{5.23}$$

$$prec(\hat{p}) = \frac{VP}{VP + FP} = \frac{30}{30 + 105} = 22\% \tag{5.24}$$

$$sens(\hat{p}) = \frac{VP}{VP + FN} = \frac{36}{36 + 15} = 91\% \tag{5.25}$$

A utilização de períodos variáveis segundo o EQM para a média móvel associada à lógica *fuzzy* para a estimação das janelas foi muito benéfica para todas as medidas de desempenho, principalmente para a taxa de erro na classe positiva, acurácia e sensibilidade. O algoritmo *T-Slow*, com essa abordagem, alcançou redução significativa na taxa de erro na classe positiva, pois caiu do patamar de aproximadamente 30% para apenas 9%.

5.4 DISCUSSÃO DA METODOLOGIA

Por se tratar de um mecanismo que visa a detecção de anomalias e não a detecção de padrões (assinatura), a metodologia proposta possui a vantagem de não se ater apenas aos ataques conhecidos, mas também aos ataques desconhecidos. Segundo Mirkovic e Reiher (2004), os mecanismos detectores de anomalia devem diligenciar sua capacidade de detectar todos os ataques contra a tendência de identificar comportamentos normais como ataques. Nesse sentido, o quadro comparativo na Tabela 5.13, fornece as análises necessárias a respeito da metodologia proposta, representada pela abordagem 4.

TAB. 5.13: Quadro Comparativo das medidas de desempenho para as Abordagens 1 à 4

Medidas de Desempenho	Janela Estática		Janela Fuzzy	
	MM30	EQM	MM30	EQM
	1	2	3	4
Acurácia	95%	88%	93%	92%
Precisão	22%	10%	19%	22%
Tx Erro Classe +	29%	27%	34%	9%
Tx Erro Classe -	5%	12%	6%	8%
Sensibilidade	71%	73%	66%	91%

O *dataset* utilizado não é rotulado, portanto, não existem informações exatas quanto ao início, duração ou término do horário do ataque. Tem-se apenas a informação que o ataque se inicia a partir de um determinado horário e que o ataque foi interrompido após o servidor tornar-se sem resposta. Infelizmente, a falta dessas informações limita a capacidade do algoritmo em fornecer resultados mais exatos para os números de falsos positivos de uma maneira geral.

Como vantagem da metodologia proposta, observa-se que a utilização de EQM nas abordagens confere maior sensibilidade ao método de detecção. A desvantagem do uso de EQM é contatada pela taxa de erro na classe negativa (TFP). Isto significa que, o algoritmo identifica mais falsos positivos com esta abordagem. Tal fato evidencia a percentagem de janelas com trafego normal que foram erradamente classificados como ataque.

Mas, como nessa metodologia privilegia-se o menor número de falsos negativos ou a taxa de erro na classe positiva (TFN), o uso da EQM juntamente com a lógica *fuzzy* é a mais relevante.

Resumindo, as vantagens da metodologia proposta nesse trabalho são as seguintes:

1. O valor da taxa de erro na classe positiva, ou a percentagem de ataques que erradamente foram qualificados como trafego normal, para a abordagem proposta na metodologia é a menor dentre as demais abordagens; e

2. Embora o valor de acurácia para a abordagem *fuzzy* EQM, tenha sido menor que as abordagens Janelas *fuzzy* MM30 e Janelas Estáticas MM30, o algoritmo T-Slow ao utilizar a abordagem Janelas *fuzzy* EQM, obteve performance significativamente melhor que as demais abordagens, dado a sua menor taxa de erro na classe positiva (TFN). Essa medida de desempenho deve ser privilegiada, pois mede a percentagem de ataques não detectados. Isto porque, um ataque não detectado pode causar

mais prejuízos ao funcionamento da rede do que o tráfego que foi incorretamente classificado como ataque.

3. Sob o ponto de vista da taxa de erro na classe positiva, as previsões analisadas pela desigualdade Tchebycheff foram mais eficazes quando se utilizou a média móvel com o menor erro quadrático médio e janelas dinâmicas pelo método da lógica *fuzzy*.

TAB. 5.14: Quadro Comparativo de Desempenho entre Trabalhos sobre a Detecção de Ataques *Slow DoS*

Resultados da Avaliação na Detecção de Ataques *Slow DoS*							
Autor	Protocolo	Método	Ataques	Sensibilidade	TFP	TFN	Acurácia
(TRIPATHI; HUBBALLI, 2018)	HTTP 2.0	Teste χ^2	*Slow Dos*	43%	0%	-	-
O presente trabalho.	HTTP 1.1	*T-Slow*	*Slow Dos*	91%	8%	9%	92%
(JAZI et al., 2017)	HTTP 1.1	CUSUM / Sampling	*Slow Dos*	100%	0%	-	-
(NAJAFABADI et al., 2016)	HTTP 1.1	*Machine Learning* (KNN e Árvore de Decisão)	R.U.D.Y.	99%	0%	-	-
(KATKAR et al., 2015)	HTTP 1.1	*Machine Learning* (Classificador Naive Bayes)	*Slow Read*	97,82%	-	-	-
(SHAFIEIAN et al., 2015)	HTTP 1.1	*Machine Learning* (*Árvores de Decisão*)	*Slow Read*	99,37%	-	-	-

5.4.1 ANÁLISE CRÍTICA

O trabalho apresentado foi direcionado à detecção dos ataques *Slow DoS* com base no histórico do comportamento do tráfego de rede.

A escolha dos atributos do tráfego de rede mais pertinentes têm papel importante na tarefa de detecção. Nesse trabalho os atributos foram escolhidos com base nas anomalias

dos ataques *Slow DoS* de forma que trabalhassem adequadamente também no mundo real. No mundo real, isto é, na análise de tráfego de rede real, acredita-se que a interferência humana é necessária, mas deve ser mínima. Segundo dos Santos (2009), o conhecimento do especialista deve ser aplicado à base de conhecimento. Isso significa que a preocupação não pode estar nas decisões na escolha de *thresholds* mais apropriados.

Quanto à detecção é importante destacar que embora os valores dos resultados para a janela fixa tenham sido um pouco melhores daqueles com janelas *fuzzy*, pode-se atribuir tal melhora ao acaso. A escolha do tamanho da janela de 30 segundos foi escolhido sem nenhuma fundamentação teórica para tal. Na literatura, alguns trabalhos utilizaram também esse mesmo tamanho de janela e outros tamanhos como 15, 20 segundos, 10 minutos, mas também sem nenhuma fundamentação teórica.

Para Zargar et al. (2013), o custo monetário pode até ser utilizado para comparar mecanismos de defesa DDoS, mas não podem servir para avaliar a eficácia do mesmo. Para a medição da eficácia do sistema de detecção, as métricas de *performance* são essenciais para a comparação entre diversos trabalhos.

A comparação desse trabalho a outros na literatura não foi feita com base na utilização do mesmo *dataset*, mas sim com base em algumas das métricas de performance segundo o modelo de detecção proposto e no intervalo de tempo utilizado.

Conforme mencionado anteriormente, Tripathi e Hubballi (2018) utilizaram janelas de tempo de 10 minutos para a detecção de ataques *Slow Rate* em um *testbed*, cujo o servidor era o Apache 2.4.23 para requisições HTTP/2. Com a utilização do teste de hipótese Qui-Quadrado, a taxa de falsos positivos foi nula, assim como a taxa de falsos negativos. No entanto, o tempo estimado para a detecção de 10 minutos não foi explicado.

De forma semelhante, Jazi et al. (2017), experimentou janelas de tempo de 15, 30, 45 e 60 segundos. Com o intervalo de tempo de 30 segundos, os autores obtiveram melhores taxas de desempenho no método proposto. O resultado para a acurácia do método foi apresentada em 100%.

Najafabadi et al. (2016) apresentou um método baseado em comitê de seleção de *features* para análise pelos classificadores de árvore de decisão e KNN, mas apenas para o ataque de conteúdo *R.U.D.Y.*. Segundo os autores, com análise de variância (ANOVA) foi possível alcançar taxa de falsos positivos de 98%.

A presente metodologia não identificou anomalias em algumas janelas onde o ataque *R.U.D.Y.* atuou. Isto porque, foi observado que o endereço IP era distinto do apresentado no gabarito segundo informações do *CIC DoS Dataset*. Tal fato pode ser explicado pela furtividade do ataque, também observada por Jazi et al. (2017), ou a uma possível

anonimização do tráfego de ataque utilizado nesse *dataset*. Na metodologia proposta, esse tipo de anomalia detectada foi considerada como falso positivo.

Katkar et al. (2015) propôs um método com base num classificador bayesiano para identificação do ataque *Slow Read*. O método conseguiu classificar o ataque *Slow Read* com acurácia de 97,82%. Se comparado ao presente trabalho, embora não se identifique o ataque *Slow Read* efetivamente e sim a anomalia a ele associada, pode-se afirmar que o *T-Slow* detectou as anomalias do *Slow Read* com 100% de acurácia.

Shafieian et al. (2015), com seu método baseado em árvores decisão, constatou que quando não é utilizada a pré-poda para o classificador árvore de decisão, a acurácia foi melhor (99,37%).

Xie e Yu (2009) analisaram ataques à camada de aplicação, com características distintas (taxas de ataque constante, crescente e estocástica), utilizando intervalo de tempo variável entre 5 e 20 segundos. Os autores escolheram a distribuição gaussiana para validar seus resultados. Ressalta-se que a escolha da distribuição gaussiana traz consigo a assunção que os resultados probabilísticos esperados tenham a conhecida forma de sino, e também, assume-se que os dados da distribuição tenham o mesmo comportamento. Assim, traz-se a reflexão que o tráfego de rede é extremamente instável e muitas das vezes os resultados fora da faixa um dado limiar não são anomalias de fato e essas ocasiões podem ser muito frequentes em função do tráfego de rede. Em tal situação, os dados não representariam uma forma de sino propriamente dita. Os autores testaram vários níveis de detecção e o nível selecionado foi $\bar{x} \pm 3s$, fornecendo a taxa de detecção de 90%.

Assim, acredita-se que a grande vantagem da metodologia proposta nessa dissertação encontra-se na taxa de detecção de falsos negativos que comparada aos demais trabalhos, resguardadas as diferenças entre *datasets* e tipos de ataques verificados, alcançou ótimos resultados.

6 CONCLUSÃO

Recentemente, como consequência dos esforços de mitigação aos ataques DoS tradicionais, os ataques *Slow DoS* surgem como ameaça à garantia da disponibilização de serviços na web. Esse ataque é considerado preocupante devido a furtividade do seu modo de operação, que lentamente e sem alarde, desabilita a vítima.

A detecção de ataques de negação de serviço possui vários desafios dentre os quais pode-se citar: obter mecanismos de detecção mais rápidos, porque esquemas de detecção consomem recursos da vítima e isso afeta a capacidade de acurácia, o *dataset* deve possuir tráfego de rede real corretamente capturado e o desempenho deverá observar baixo índice de falsos positivos e falsos negativos e altos índices de verdadeiros positivos e verdadeiros negativos.

Uma das formas de verificação de normalidade de uma distribuição é o cálculo das distribuições amostrais. Uma das vantagens da metodologia proposta é que não há um *threshold* previamente definido. A verificação de normalidade da distribuição de pacotes é feita mediante a observação dos valores prévios ou analogia histórica do próprio tráfego que servirá para alertar sobre anomalias sem limiares previamente arbitrados, tais como limitações do número de conexões por IP.

Entre algumas das contribuições dessa dissertação, destacam-se as seguintes:

- A proposição de um algoritmo e métricas para a detecção de anomalias de ataques de negação de serviço;
- A apresentação de uma discussão sobre quais medidas de desempenho devem ser privilegiadas na detecção de anomalias no tráfego de rede;
- Apresentação de métodos capazes de prover um dimensionamento mais adequado das janelas de tempo; e
- Apresentação de um método para a amostragem do tráfego de rede com menor custo de processamento.

A grande expectativa desse trabalho consistia na capacidade de desempenho alcançado pelo algoritmo *T-Slow* segundo a metodologia proposta. Nesse sentido, as previsões analisadas pela desigualdade Tchebycheff foram mais eficazes quando utilizou-se a média

móvel com o menor erro quadrático médio e janelas dinâmicas pelo método da lógica *fuzzy*.

Se fossem analisados os resultados das métricas F.O.R. mediante o grau de invisibilidade dos ataques lentos com relação as mesmas, poderia-se citar que a primeira ocorrência do ataque *Slow Body* implementado pela ferramenta *R.U.D.Y.* foi o ataque que se mostrou impossível de ser detectado pelas métricas em todas as estrátégias apresentadas. Tal fato corrobora para a afirmação da característica furtiva do ataque para a tarefa de detecção.

Antes da aplicação dos testes destinados às distribuições normais, é aconselhável a verificação dos critérios necessários para determinar se a distribuição supostamente normal, de fato, é normal. A análise de distribuição permite verificar se determinado conjunto de dados ajusta-se ou não à uma distribuição normal. Assim, a decisão sobre como analisar os dados será mais correta. Com a ferramenta Alteryx (HALPER, 2014), utilizada nesse trabalho, um dos critérios verificados foi o suficiente para desencorajar a utilização da distribuição normal, conforme Apêndice 1 desse trabalho. O valor de significância da estatística qui-quadrado (χ^2) foi utilizado para verificar se distribuição do tráfego em questão, possui um bom ajuste à distribuição normal. O valor do nível de significância utilizado (α) é o de 0,05 e segundo Navidi (2012) essa é uma escolha comum. O valor calculado para esse indicador foi zero, logo rejeitou-se a plausibilidade do ajuste dos dados à distribuição normal.

A desigualdade Tchebycheff oferece uma boa oportunidade de análise dos dados do tráfego de rede, pois aplica-se a qualquer distribuição. Apenas recordando que com a desigualdade Tchebycheff estima-se que pelo menos 89% do tráfego considerado normal ficará dentro dos limiares de $\pm 3s$ a partir da média $\bar{X}$, diferentemente do estipulado para o limite das distribuições que ao utilizar $\pm 3s$, estima-se que 99,7% dos valores estarão distribuídos a partir da média.

Não apenas as métricas escolhidas são importantes na tarefa de detecção. Procurou-se evidenciar outros fatores tão importantes quanto as métricas para o sucesso do sistema de detecção, tais como o tamanho período analisado ou tamanho da amostra, o intervalo de tempo considerado e a estatística-teste para análise de detecção. Todos esses parâmetros fazem parte da estratégia de detecção e foram apresentados nesse trabalho a partir de critérios bem definidos.

Comparando-se esse trabalho a outros mecanismos de detecção, é difícil a realização de uma comparação mais extensa entre os diversos mecanismos, porque muitos bons trabalhos (HONG et al., 2017; HIRAKAWA et al., 2016) não apresentam as métricas necessárias para a avaliação da eficácia do método.

Optou-se por utilizar a matriz confusão, para analisar e obter medidas de desempenho, por ser um problema com duas classes. A classe positiva representa a presença de anomalia em função dos ataques lentos e a classe negativa, a ausência desse tipo de ataque. A partir dos dados da matriz confusão, foram calculadas as medidas de desempenho.

Vislumbra-se a possibilidade de aprimoramento nos resultados apresentados nesse trabalho. Então, como trabalhos futuros, elencam-se as seguintes possibilidades:

- Investigar se a escolha de outras variáveis linguísticas para a lógica *fuzzy* são capazes de melhorar o desempenho do algoritmo no que tange à sua acurácia;
- Propor um algoritmo de identificação de ataques *Slow DoS*. Nesse caso, além da detecção da anomalia será possível fornecer a nomenclatura do ataque; e
- A realimentação do sistema de detecção com possíveis ajustes na lógica *fuzzy*.

A lógica *fuzzy* tornou possível estabelecer o tamanho das janelas de tempo em função das variáveis escolhidas e segundo diferentes conceitos ou regras de produção. Optou-se pela lógica *fuzzy* principalmente porque o ambiente de rede é um ambiente instável.

Como trabalho futuro, pretende-se testar, como variáveis linguísticas da lógica *fuzzy*, outras variáveis ou atributos do tráfego de rede que também são afetadas pela ocorrência de anomalias. Pretende-se implantar a realimentação do sistema de detecção, fornecendo informações a um especialista. Caso a conclusão final do mecanismo de detecção seja negativa para ataque, o tráfego seguirá o seu fluxo normal e nenhuma ação de mitigação será demandada. Mas, se houver detecção de anomalia, o tráfego anormal será encaminhado para análise de um especialista. O especialista poderá efetuar correções ou ajustes nas regras de fuzzificação ou defuzzificação, por exemplo.

Para auxiliar nessa tarefa existem algumas técnicas da análise preditiva que podem ser consideradas, como por exemplo o algoritmo de classificação Árvore de Decisão para a escolha dos atributos mais significativos, geralmente utilizado em um *dataset* devidamente rotulado em tráfego normal e tráfego de ataque (anormal).

A classificação automática dos ataques pode ser realizada utilizando algoritmos de aprendizado de máquina. Os classificadores também podem ser úteis na detecção do tráfego de ataque, como por exemplo, na formação de um comitê de classificadores. Deseja-se investigar se um comitê de classificadores com votação uniforme, por exemplo, pode efetivamente melhorar a predição dos ataques lentos à camada de aplicação. As saidas de classificadores distintos, cada um com diferentes viéses, poderia ser utilizada para esse fim.

Há grande utilidade em se trabalhar com classificadores com viéses distintos. Segundo (FACELI et al., 2015), a ideia por trás de qualquer modelo múltiplo preditivo é baseada na observação de que diferentes algoritmos de aprendizado exploram: diferentes linguagens linguagens de representação e diferentes funções de avaliação de hipóteses. Por exemplo, o algoritmo Naive Bayes cria um modelo binomial ou multinomial de classificação probabilística da relação entre um conjunto de variáveis preditoras e uma variável alvo categórica. Já o algoritmo utilizado para criar as regras de decisão da árvore de decisão utiliza a estratégia dividir para conquistar.

Essa abordagem é interessante, porque o processo de customização de uma árvore de decisão envolve a configuração de alguns parâmetros. Os parâmetros mais importantes são:

a) a escolha do algoritmo que melhor selecionará as partições ou *splits*, que dirão de que forma o algoritmo irá dividir o conjunto de dados; e

b) a escolha da estratégia de poda para impedir o problema da super generalização dos dados de treinamento chamado *overfitting*.

Conforme já mencionado, pretende-se propor um sistema de defesa que além de detectar, também seja capaz de identificar os ataques lentos à camada de aplicação pela adição das métricas de identificação.

A tarefa de identificação é mais difícil que a detecção, porque com base na assinatura do ataque, provê nome e endereço do invasor e por esse motivo, é de grande utilidade aos administradores para a mitigação dos danos causados aos servidores. Como os administradores são a primeira linha de defesa, como trabalho futuro, planeja-se um sistema de defesa mais completo, que possa fornecer mais detalhes sobre a rede, para futuros ajustes no próprio sistema ou tomada de decisão.

De acordo com Moustis e Kotzanikolaou (2013), embora *firewalls* ou proteção IPS possam impedir efetivamente um invasor, essas soluções ainda podem levar a muitos falsos positivos. Ataques efetuados em nível de aplicação são difíceis de detectar mediante o uso de *firewalls*. O processo de filtragem de pacotes neste nível é computacionalmente caro e difícil de ser escalável, e ainda, pode gerar muitos falsos positivos, bloqueando assim, clientes legítimos (MICHELIN, 2015).

São muitos os desafios e recentemente, como consequência dos esforços de mitigação aos ataques DoS tradicionais, os ataques Slow DoS surgem como ameaça à garantia da disponibilização de serviços na web.

Espera-se que esse trabalho seja útil a outros pesquisadores, seja pela utilização das métricas aqui propostas em outros sistemas ou a utilização do sistema proposto com outros *datasets*. Enfim, que efetivamente possa ser comparado e discutido à luz dos requisisitos necessários para a tarefa de detecção de anomalias ou identificação de ataques.

7 REFERÊNCIAS BIBLIOGRÁFICAS

AKAMAI. Q1 report 2017 akamai state of the internet. **Q1 Report 2017 Akamai State of the Internet**, v. Q1 2017, 2017. Disponível em: <https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q1-2017-state-of-the-internet-connectivity-report.pdf/>. Acesso em: 21 jun. de 2017.

AMEYED, D.; JAAFAR, F. ; FATTAHI, J. A slow read attack using cloud. In: 2015 7TH INTERNATIONAL CONFERENCE ON ELECTRONICS, COMPUTERS AND ARTIFICIAL INTELLIGENCE (ECAI), 7., 2015. **Anais eletrônicos**. Romania: IEEE, 2015, p. SSS–33–SSS–38. Disponível em: <https://ieeexplore.ieee.org/document/7301202/>. Acesso em: 31 nov. de 2015.

AQIL, A.; ATYA, A. O.; JAEGER, T.; KRISHNAMURTHY, S. V.; LEVITT, K.; MCDANIEL, P. D.; ROWE, J. ; SWAMI, A. Detection of stealthy tcp-based dos attacks. In: MILITARY COMMUNICATIONS CONFERENCE, MILCOM 2015-2015 IEEE, [s.n]., 2015. **Anais eletrônicos**. [S.l.: s.n.], 2015, p. 348–353. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7357467/>. Acesso em: 14 set. de 2017.

BHUYAN, M. H.; KASHYAP, H. J.; BHATTACHARYYA, D. K. ; KALITA, J. K. Detecting distributed denial of service attacks: methods, tools and future directions. **The Computer Journal**, v. 57, n. 4, p. 537–556, 2014. Disponível em: <https://academic.oup.com/comjnl/article/57/4/537/407932>. Acesso em: 21 jan. de 2018.

CAMBIASO, E.; PAPALEO, G. ; AIELLO, M. Taxonomy of slow dos attacks to web applications. In: INTERNATIONAL CONFERENCE ON SECURITY IN COMPUTER NETWORKS AND DISTRIBUTED SYSTEMS, [s.n.]., 2012. **Anais eletrônicos**. [S.l.: s.n.], 2012, p. 195–204. Disponível em: <https://link.springer.com/chapter/10.1007em: 22 abr. de 2017.

CAMBIASO, E.; PAPALEO, G. ; AIELLO, M. Slowcomm: Design, development and performance evaluation of a new slow dos attack. **Journal of Information Security and Applications**, v. 35, p. 23–31, 2017. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2214212616300680>. Acesso em: 21 jan. de 2018.

CAMBIASO, E.; PAPALEO, G.; CHIOLA, G. ; AIELLO, M. Mobile executions of slow dos attacks. **Logic Journal of the IGPL**, v. 24, n. 1, p. 54–67, 2016. Disponível em: <http://dx.doi.org/10.1093/jigpal/jzv043>. Acesso em: 15 jan. de 2017.

CERT-U.S. Ddos. **National Cybersecurity and Communications Integration Center**, v. 2014, 2014. Disponível em: <https://www.us-cert.gov/sites/default/files/publications/DDoSAcesso em: nov. de 2016.

CGI.BR. Cartilha de segurança para internet. **CGI. br (Comitê Gestor da Internet no Brasil)**, v. 2012, 2012. Disponível em: <https://www.cgi.br/>. Acesso em: nov. de 2016.

CGI.BR. Cartilha de segurança para internet. **CGI. br (Comitê Gestor da Internet no Brasil)**, v. 2012, 2012. Disponível em: <https://www.cgi.br/>. Acesso em: nov. de 2016.

CHEE, W. O.; BRENNAN, T. Http post slide show. In: [S.N.], [s.n.]., 2010. **Anais eletrônicos**. [S.l.: s.n.], 2010, p. 1. Disponível em: <https://www.owasp.org/index.php/OWASPHTTPPostTool>. Acesso em: 21 jan. de 2018.

CIC-DOS. **University of New Brunswick-Canadian Institute for Cybersecurity (CIC)-DoS Dataset**. [S.l.: s.n.], 2017. (Relatório Técnico).

DAMON, E.; DALE, J.; LARON, E.; MACHE, J.; LAND, N. ; WEISS, R. Hands-on denial of service lab exercises using slowloris and rudy. In: PROCEEDINGS OF THE 2012 INFORMATION SECURITY CURRICULUM DEVELOPMENT CONFERENCE, [s.n]., 2012. **Anais eletrônicos**. [S.l.: s.n.], 2012, p. 21–29. Disponível em: <https://dl.acm.org/citation.cfm?id=2390321>. Acesso em: 14 set. de 2017.

DANTAS, Y. G.; NIGAM, V. ; FONSECA, I. E. A selective defense for application layer ddos attacks. In: INTELLIGENCE AND SECURITY INFORMATICS CONFERENCE (JISIC), 2014 IEEE JOINT, [s.n]., 2014. **Anais eletrônicos**. [S.l.: s.n.], 2014, p. 75–82. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7155851/>. Acesso em: 14 set. de 2017.

DOS SANTOS, A. F. P. **Identificação e Análise de Comportamentos Anômalos**. 2009. 106 f. Tese (Doutorado em Modelagem Computacional) – Laboratório Nacional de Computação Científica, Petrópolis, 2009.

FACELI, K.; LORENA, A. C.; GAMA, J. ; CARVALHO, A. C. **Inteligência Artificial: Uma abordagem de aprendizagem de máquina**. [S.l.]: Grupo Gen-LTC, 2015.

GONZALEZ, H.; GOSSELIN-LAVIGNE, M. A.; GHORBANI, A.; ISSAC, B. ; ISRAR, N. **The impact of application layer denial of service attacks**. [S.l.]: CRC Press, 2014.

GUPTA, B.; BADVE, O. P. Taxonomy of dos and ddos attacks and desirable defense mechanism in a cloud computing environment. **Neural Computing and Applications**, v. 28, n. 12, p. 3655–3682, 2017. Disponível em: <https://link.springer.com/journal/521/28/12/page/1>. Acesso em: 21 out. de 2017.

HALPER, F. Predictive analytics for business advantage. **TDWI Research**, v. [s.n.], 2014. Disponível em: <http://pages.alteryx.com/rs/alteryx/images/TDWI-BestPracticesReport_Predictive-Analytics.pdf>. Acesso em: 06 mai. de 2017.

HELALAT, S. M. An investigation of the impact of the slow http dos and ddos attacks on the cloud environment. In: [S.N.], [s.n]., 2017. **Anais eletrônicos**. [S.l.: s.n.], 2017, p.

74. Disponível em: <http://www.diva-portal.org/smash/record.jsf?pid=diva2Acesso em: 14 set. de 2017.

HIRAKAWA, T.; OGURA, K.; BISTA, B. B. ; TAKATA, T. A defense method against distributed slow http dos attack. In: NETWORK-BASED INFORMATION SYSTEMS (NBIS), 2016 19TH INTERNATIONAL CONFERENCE ON, [s.n]., 2016. **Anais eletrônicos**. [S.l.: s.n.], 2016, p. 152–158. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7789752/>. Acesso em: 14 nov. de 2017.

HONG, K.; KIM, Y.; CHOI, H. ; PARK, J. Sdn-assisted slow http ddos attack defense method. **IEEE Communications Letters**, v. [s.n.], p. [s.n.], 2017. Disponível em: <https://ieeexplore.ieee.org/abstract/document/1378371/>. Acesso em: 21 jan. de 2018.

JANTZEN, J. Design of fuzzy controllers. **Technical University of Denmark, Department of Automation, Bldg**, v. 326, p. 362–367, 1998. Disponível em: <https://tinyurl.com/ybhzaxkn>. Acesso em: 30 abr. de 2018.

JAZI, H. H.; GONZALEZ, H.; STAKHANOVA, N. ; GHORBANI, A. A. Detecting http-based application layer dos attacks on web servers in the presence of sampling. **Computer Networks**, v. 121, p. 25–36, 2017. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128617301172>. Acesso em: 14 set. de 2017.

JIANG, M.; WANG, C.; LUO, X.; MIU, M. ; CHEN, T. Characterizing the impacts of application layer ddos attacks. In: WEB SERVICES (ICWS), 2017 IEEE INTERNATIONAL CONFERENCE ON, [s.n]., 2017. **Anais eletrônicos**. [S.l.: s.n.], 2017, p. 500–507. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8029800/>. Acesso em: 15 set. de 2017.

KATKAR, V.; ZINJADE, A.; DALVI, S.; BAFNA, T. ; MAHAJAN, R. Detection of dos/ddos attack against http servers using naive bayesian. In: COMPUTING COMMUNICATION CONTROL AND AUTOMATION (ICCUBEA), 2015 INTERNATIONAL CONFERENCE ON, [s.n]., 2015. **Anais eletrônicos**. [S.l.: s.n.], 2015, p. 280–285. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7155851/>. Acesso em: 14 set. de 2017.

KHANNA, S.; VENKATESH, S. S.; FATEMIEH, O.; KHAN, F. ; GUNTER, C. A. Adaptive selective verification: An efficient adaptive countermeasure to thwart dos attacks. **IEEE/ACM Transactions on Networking (TON)**, v. 20, n. 3, p. 715–728, 2012. Disponível em: <https://dl.acm.org/citation.cfm?id=2369163>. Acesso em: 24 fev. de 2018.

KUROSE, J. F.; ROSS, K. W. Redes de computadores e a internet. **Uma nova**, v. [s.n], p. [s.n.], 2006.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet**. 6. ed. Rio de Janeiro: Pearson, 2014. 634 p.

LOPES, P. A. **Probabilidades & estatística**. [S.l.]: Reichmann & Affonso, 2001.

MAMDANI, E. H.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. **International journal of man-machine studies**, v. 7, n. 1, p. 1–13, 1975. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020737375800022>. Acesso em: 30 abr. de 2018.

MENASCÉ, D. A.; ALMEIDA, V. A. Planejamento de capacidade para serviços na web: Métricas, modelos e métodos. **Rio de Janeiro: Campus**, v. [s.n], p. [s.n.], 2003.

MICHELIN, R. A. **Mitigação de ataques de negação de serviço em Rests autenticáveis na nuvem**. 2015. 134 f. Dissertação (Mestrado em Sistemas e Computação) – Pontifícia Universidade Católica do Rio Grande do Sul, Rio Grande do Sul, 2015. Disponível em: <http://tede2.pucrs.br/tede2/handle/tede/5272>. Acesso em: 10 mar. de 2018.

MIRKOVIC, J.; REIHER, P. A taxonomy of ddos attack and ddos defense mechanisms. **ACM SIGCOMM Computer Communication Review**, v. 34, n. 2, p. 39–53, 2004. Disponível em: <https://dl.acm.org/citation.cfm?id=997156>. Acesso em: 21 jan. de 2018.

MOUSTIS, D.; KOTZANIKOLAOU, P. Evaluating security controls against http-based ddos attacks. In: INFORMATION, INTELLIGENCE, SYSTEMS AND APPLICATIONS (IISA), 2013 FOURTH INTERNATIONAL CONFERENCE ON, [s.n.]., 2013. **Anais eletrônicos**. [S.l.: s.n.], 2013, p. 1–6. Disponível em: <https://ieeexplore.ieee.org/abstract/document/6623707/>. Acesso em: 14 set. de 2018.

NAJAFABADI, M. M.; KHOSHGOFTAAR, T. M.; NAPOLITANO, A. ; WHEELUS, C. Rudy attack: Detection at the network level and its important features. In: FLAIRS CONFERENCE, [s.n.]., 2016. **Anais eletrônicos**. [S.l.: s.n.], 2016, p. 288–293. Disponível em: <https://tinyurl.com/yal3j6rd>. Acesso em: 21 jan. de 2018.

NAVIDI, W. **Probabilidade e estatística para ciências exatas**. [S.l.]: AMGH Editora, 2012.

OHSITA, Y.; ATA, S. ; MURATA, M. Detecting distributed denial-of-service attacks by analyzing tcp syn packets statistically. **IEICE transactions on communications**, v. 89, n. 10, p. 2868–2877, 2006. Disponível em: <https://ieeexplore.ieee.org/abstract/document/1378371/>. Acesso em: 21 jan. de 2018.

OSANAIYE, O.; CHOO, K.-K. R. ; DLODLO, M. Distributed denial of service (ddos) resilience in cloud: review and conceptual cloud ddos mitigation framework. **Journal of Network and Computer Applications**, v. 67, p. 147–165, 2016. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804516000023>. Acesso em: 21 jan. de 2018.

PARK, J.; IWAI, K.; TANAKA, H. ; KUROKAWA, T. Analysis of slow read dos attack. In: INFORMATION THEORY AND ITS APPLICATIONS (ISITA), 2014 INTERNATIONAL SYMPOSIUM ON, [s.n]., 2014. **Anais eletrônicos**. [S.l.: s.n.], 2014, p.

60–64. Disponível em: <https://ieeexplore.ieee.org/abstract/document/6979803/>. Acesso em: 14 set. de 2017.

PENG, T.; LECKIE, C. ; RAMAMOHANARAO, K. Survey of network-based defense mechanisms countering the dos and ddos problems. **ACM Computing Surveys (CSUR)**, v. 39, n. 1, p. 3, 2007. Disponível em: <https://dl.acm.org/citation.cfm?id=1216373>. Acesso em: 20 set. de 2017.

RAZ, R. Universal http dos-are you dead yet. **http://chaptersinwebsecurity.blogspot.ca**, v. 1, p. [s.n.], 2010. Disponível em: <http://chaptersinwebsecurity.blogspot.ca/2010>. Acesso em: 21 jan. de 2018.

REZENDE, S. O. **Sistemas inteligentes: fundamentos e aplicações**. [S.l.]: Editora Manole Ltda, 2003.

RILEY, G. F.; HENDERSON, T. R. The ns-3 network simulator. In: MODELING AND TOOLS FOR NETWORK SIMULATION, [s.n.]., 2010. **Anais eletrônicos**. [S.l.]: Springer, 2010, p. 15–34. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-642-12331-3_2>. Acesso em: 30 abr. de 2018.

ROSS, T. J. **Fuzzy logic with engineering applications**. [S.l.]: John Wiley & Sons, 2009.

RSNAKE, J. K.; LEE, R. Slowloris http dos. **URL: http://ha.ckers. org/slowloris/(June 2009)**, v. 1, p. s/n, 2009. Disponível em: <http://ha.ckers.org/slowloris/(June 2009)>. Acesso em: 21 jan. de 2018.

SAVAGE, I. R. Probability inequalities of the tchebycheff type. **Journal of Research of the National Bureau of Standards-B. Mathematics and Mathematical Physics B**, v. 65, n. 3, p. 211–222, 1961. Disponível em: <https://nvlpubs.nist.gov/nistpubs/jres/65B/jresv65Bn3p211-A1b.pdf>. Acesso em: 21 jan. de 2018.

SHAFIEIAN, S.; ZULKERNINE, M. ; HAQUE, A. Cloudzombie: Launching and detecting slow-read distributed denial of service attacks from the cloud. In: COMPUTER AND INFORMATION TECHNOLOGY; UBIQUITOUS COMPUTING AND COMMUNICATIONS; DEPENDABLE, AUTONOMIC AND SECURE COMPUTING; PERVASIVE INTELLIGENCE AND COMPUTING (CIT/IUCC/DASC/PICOM), 2015 IEEE INTERNATIONAL CONFERENCE ON, [s.n.]., 2015. **Anais eletrônicos**. [S.l.: s.n.], 2015, p. 1733–1740. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7363307/>. Acesso em: 30 abr. de 2018.

SHAMELI-SENDI, A.; POURZANDI, M.; FEKIH-AHMED, M. ; CHERIET, M. Taxonomy of distributed denial of service mitigation approaches for cloud computing. **Journal of Network and Computer Applications**, v. 58, p. 165–179, 2015. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804515002118>. Acesso em: 21 jan. de 2018.

SHARAFALDIN, I.; GHARIB, A.; LASHKARI, A. H. ; GHORBANI, A. A. Towards a reliable intrusion detection benchmark dataset. **Software Networking**, v. 2018, n. 1, p. 177–200, 2018. Disponível em: <http://www.riverpublishers.com/journalreadhtmlarticle.php?j=JSN/2017/1/9>. Acesso em: 21 jan. de 2018.

SHEKYAN, S. Slowhttptest-application layer dos attack simulator. **línea]. Available: http://code. google. com/p/slowhttptest**, v. s/n, p. 1, 2013. Disponível em: <http://code.google.com/p/slowhttptest>. Acesso em: 21 jan. de 2018.

SHIRAVI, A.; SHIRAVI, H.; TAVALLAEE, M. ; GHORBANI, A. A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. **computers & security**, v. 31, n. 3, p. 357–374, 2012. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167404811001672>. Acesso em: 21 jan. de 2018.

SILVA, S. S.; SILVA, R. M.; PINTO, R. C. ; SALLES, R. M. Botnets: A survey. **Computer Networks**, v. 57, n. 2, p. 378–403, 2013. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1389128612003568>. Acesso em: 24 fev. de 2018.

SIMULATOR, Q. N. Scalable network technologies. **Inc.[Online]. Available: www.qualnet.com**, v. [s.n.], 2011. Disponível em: <https://web.scalable-networks.com/qualnet-network-simulator-software>. Acesso em: 30 abr. de 2018.

TRIPATHI, N.; HUBBALLI, N. Slow rate denial of service attacks against http/2 and detection. **Computers & Security**, v. 72, p. 255–272, 2018. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167404817301980>. Acesso em: 25 mar. de 2018.

VARGA, A.; HORNIG, R. An overview of the omnet++ simulation environment. In: PROCEEDINGS OF THE 1ST INTERNATIONAL CONFERENCE ON SIMULATION TOOLS AND TECHNIQUES FOR COMMUNICATIONS, NETWORKS AND SYSTEMS & WORKSHOPS, [s.n.]., 2008. **Anais eletrônicos**. [S.l.: s.n.], 2008, p. 60. Disponível em: <https://dl.acm.org/citation.cfm?id=1416290>. Acesso em: 30 abr. de 2018.

WHEELUS, C.; KHOSHGOFTAAR, T. M.; ZUECH, R. ; NAJAFABADI, M. M. A session based approach for aggregating network traffic data–the santa dataset. In: BIOINFORMATICS AND BIOENGINEERING (BIBE), 2014 IEEE INTERNATIONAL CONFERENCE ON, s/n., 2014. **Anais eletrônicos**. [S.l.: s.n.], 2014, p. 369–378. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7033608/>. Acesso em: 21 jan. de 2018.

XIE, Y.; YU, S.-Z. Monitoring the application-layer ddos attacks for popular websites. **IEEE/ACM Transactions on Networking (TON)**, v. 17, n. 1, p. 15–25, 2009. Disponível em: <https://dl.acm.org/citation.cfm?id=1514072>. Acesso em: 14 set. de 2018.

YAN, Q.; YU, F. R.; GONG, Q. ; LI, J. Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges. **IEEE Communications Surveys & Tutorials**, v. 18, n. 1, p. 602–622, 2016. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7289347/>. Acesso em: 30 abr. de 2018.

ZADEH, L. A. A fuzzy-algorithmic approach to the definition of complex or imprecise concepts. **Systems Theory in the Social Sciences**, v. [s.n.], p. 202–282, 1976. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-0348-5495-5-11>. Acesso em: 21 jan. de 2018.

ZARGAR, S. T.; JOSHI, J. ; TIPPER, D. A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. **IEEE communications surveys & tutorials**, v. 15, n. 4, p. 2046–2069, 2013. Disponível em: <https://ieeexplore.ieee.org/abstract/document/6489876/>. Acesso em: 25 mar. de 2018.

8 APÊNDICES

APÊNDICE 1: ANÁLISE DE DISTRIBUIÇÃO

Os valores de uma distribuição normal raramente contém valores atípicos (NAVIDI, 2012).

Uma distribuição é dita normal quando seus valores adequam-se ou ajustam-se à função densidade de probabilidade definida pela Equação 8.1.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{8.1}$$

Os dados da variável RTTR não se ajustam à uma distribuição normal, conforme Figura 8.1.

E, da mesma forma procedeu-se a análise da distribuição dos dados para os valores da métrica *Flags*, conforme Figura 8.2.

Segundo Navidi (2012), para amostras com valores atípicos à direita, transformamos os dados aplicando o logaritmo natural (ou qualquer logaritmo) em cada um dos valores. Em seguida, procura-se determinar se os logaritmos constituem uma população normal traçando o gráfico deles na forma de histograma ou de um gráfico de probabilidade.

A aplicação de uma estatística teste com ajuste de dados segundo uma Distribuição Log-Normal foi desconsiderada em função dos valores negativos e zeros presentes entre os valores da variável *Flags_ Prop*.

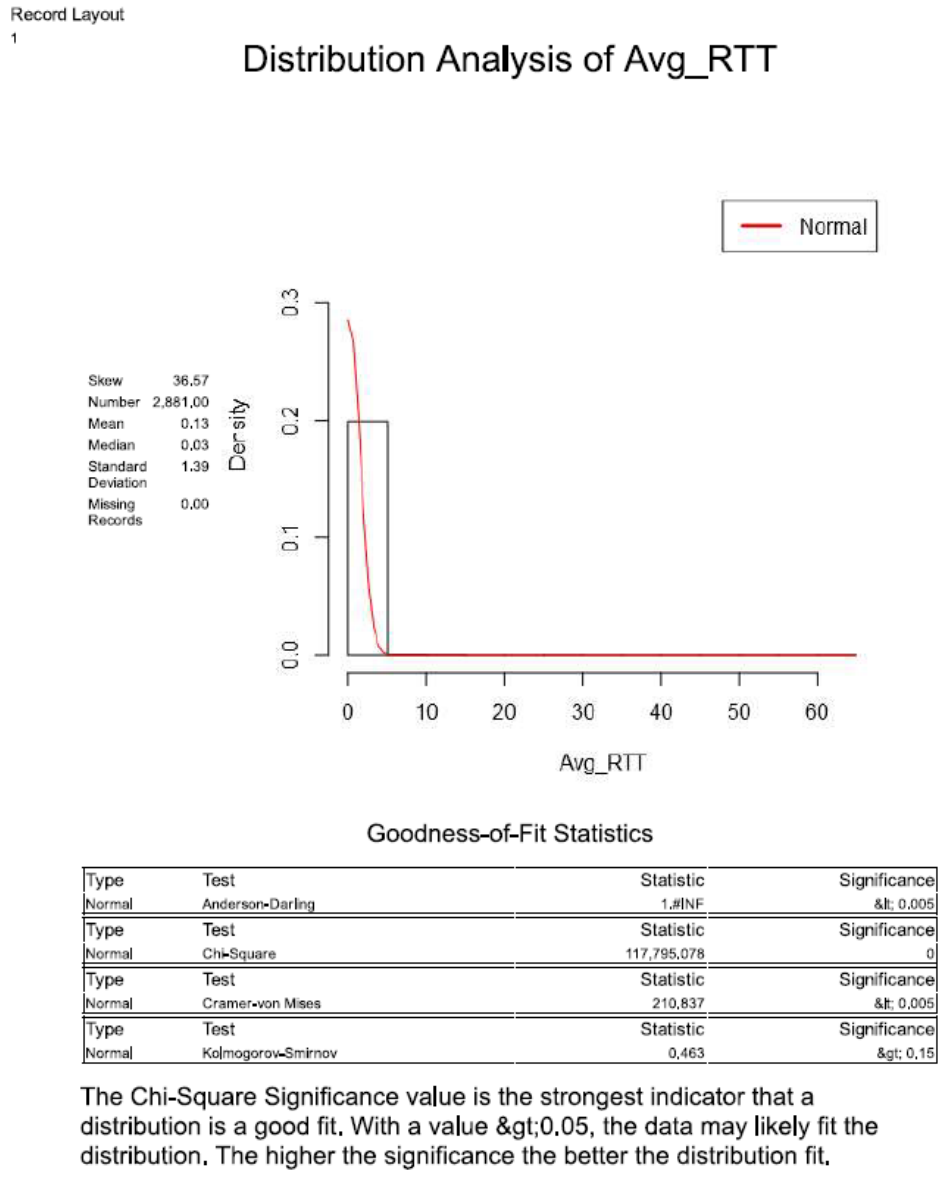
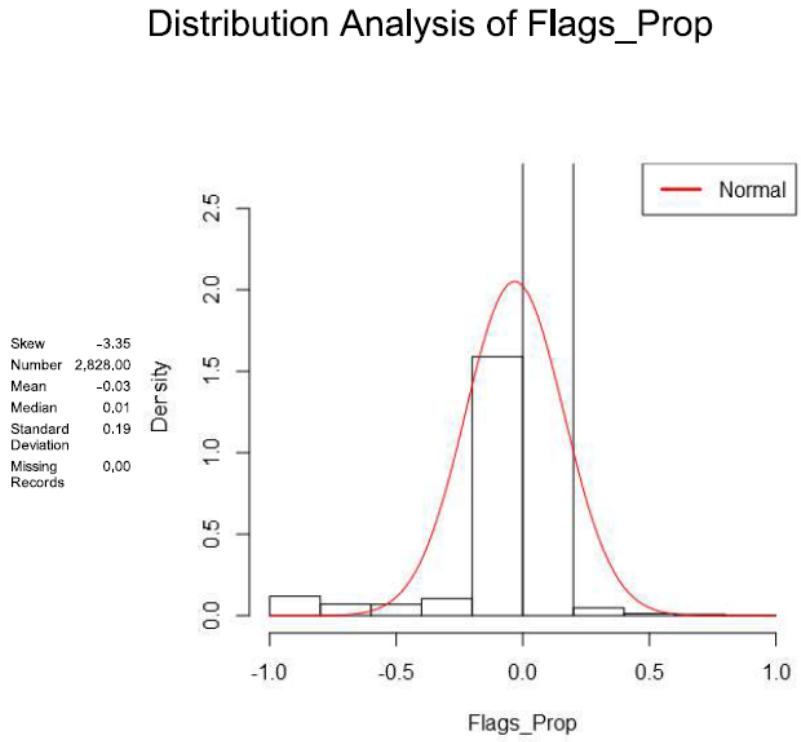


FIG. 8.1: Gráfico de ajustamento à Distribuição Normal para a distribuição da média RTTR por janela de tempo estática.

Record Layout
1



Goodness-of-Fit Statistics

Type	Test	Statistic	Significance
Normal	Anderson-Darling	574.611	< 0.005
Type	Test	Statistic	Significance
Normal	Chi-Square	40,615,667	0
Type	Test	Statistic	Significance
Normal	Cramer-von Mises	117.691	< 0.005
Type	Test	Statistic	Significance
Normal	Kolmogorov-Smirnov	0.357	> 0.15

FIG. 8.2: Gráfico de ajustamento à Distribuição Normal para a distribuição de Flags_Prop por janela de tempo estática.

9 ANEXOS

ANEXO 1: PROVA DA DESIGUALDADE TCHEBYCHEFF

Nos casos onde se conhece o valor médio e o desvio-padrão da variável aleatória, mas a respectiva distribuição não é conhecida, fato que tornaria impossível o cálculo da distribuição de probabilidades dessa variável, utiliza-se a desigualdade Tchebycheff. O conhecimento de posição e escala de dispersão permitem fazer avaliações de probabilidades.

Para uma variável aleatória X com esperança matemática dada por $\mu = E(X)$ e desvio padrão dado por $\sigma^2 = V(X)$ a desigualdade Tchebycheff é definida por:

$$P(|x-\mu|\sigma) \geq k\sigma) \leq \frac{1}{k^2} \tag{9.1}$$

Segue a demonstração desse importante teorema proposto pelo famoso matemático russo Pafnuty Tchebycheff.

No caso de uma variável aleatória contínua X:

$$\sigma^2 = \int_{-\infty}^{+\infty} (x-\mu)^2 f(x)\, dx \tag{9.2}$$

$$= \int_{-\infty}^{\mu-k\sigma} (x-\mu)^2 f(x)\, dx + \int_{\mu-k\sigma}^{\mu+k\sigma} (x-\mu)^2 f(x)\, dx + \int_{\mu+k\sigma}^{\infty} (x-\mu)^2 f(x)\, dx \tag{9.3}$$

Analisando o primeiro e terceiro termo tem-se que:

$$\sigma^2 \geq \int_{-\infty}^{\mu-k\sigma} (x-\mu)^2 f(x)\, dx + \int_{\mu+k\sigma}^{\infty} (x-\mu)^2 f(x)\, dx \tag{9.4}$$

Na primeira integral, para os valores de x_j no domínio de $|x-\mu| \geq k\sigma$ tem-se que:

$$x \leq \mu - k\sigma \tag{9.5}$$

$$k\sigma \leq \mu - x \tag{9.6}$$

$$-k\sigma \geq (x-\mu) \tag{9.7}$$

$$(x-\mu)^2 \geq k^2\sigma^2 \tag{9.8}$$

Da mesma forma, na segunda integral, para os valores de x_j no domínio de $|x-\mu| \geq k\sigma$

tem-se que:

$$x \geq \mu + k\sigma \tag{9.9}$$

$$(x - \mu) \geq k\sigma \tag{9.10}$$

$$(x - \mu)^2 \geq k^2\sigma^2 \tag{9.11}$$

Substituindo em (9.4) obtemos a seguinte desigualdade:

$$\sigma^2 \geq \int_{-\infty}^{\mu - k\sigma} k^2\sigma^2 f(x)\, dx + \int_{\mu + k\sigma}^{\infty} k^2\sigma^2 f(x)\, dx \tag{9.12}$$

E, dividindo todos os termos por k^2, obtemos a prova da desigualdade Tchebycheff.

$$\frac{1}{k^2} \geq \int_{-\infty}^{\mu - k\sigma} f(x)\, dx + \int_{\mu + k\sigma}^{\infty} f(x)\, dx \tag{9.13}$$

$$\frac{1}{k^2} \geq P[X \leq \mu - k\sigma] + P[X \geq \mu + k\sigma] \tag{9.14}$$

$$\frac{1}{k^2} \geq P[X - \mu \leq -k\sigma] + P[X - \mu \geq k\sigma] \tag{9.15}$$

$$P[|X - \mu| \geq k\sigma] \leq \frac{1}{k^2} \tag{9.16}$$