UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de São José do Rio Preto

Vinícius Oliveira Ferreira

Classificação de anomalias e redução de falsos positivos em sistemas de detecção de intrusão baseados em rede utilizando métodos de agrupamento

São José do Rio Preto
2016

Vinícius Oliveira Ferreira

Classificação de anomalias e redução de falsos positivos em sistemas de detecção de intrusão baseados em rede utilizando métodos de agrupamento

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de São José do Rio Preto.

Orientador: Prof. Dr. Adriano Mauro Cansian

São José do Rio Preto
2016

Ferreira, Vinícius Oliveira.
Classificação de anomalias e redução de falsos positivos em sistemas de detecção de intrusão baseados em rede utilizando métodos de agrupamento / Vinícius Oliveira Ferreira. -- São José do Rio Preto, 2016
90 f. : il., tabs.

Orientador: Adriano Mauro Cansian
Dissertação (mestrado) – Universidade Estadual Paulista “Júlio de Mesquita Filho”, Instituto de Biociências, Letras e Ciências Exatas

1. Computação. 2. Redes de computadores. 3. Sistemas de detecção de intrusão (Medidas de segurança) 4. Algoritmos de computador. I. Cansian, Adriano Mauro. II. Universidade Estadual Paulista "Júlio de Mesquita Filho". Instituto de Biociências, Letras e Ciências Exatas. III. Título.

CDU – 681.3.025

Ficha catalográfica elaborada pela Biblioteca do IBILCE
UNESP - Câmpus de São José do Rio Preto

Vinícius Oliveira Ferreira

Classificação de anomalias e redução de falsos positivos em sistemas de detecção de intrusão baseados em rede utilizando métodos de agrupamento

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de São José do Rio Preto.

Comissão Examinadora

Prof. Dr. Adriano Mauro Cansian
UNESP – São José do Rio Preto
Orientador

Prof. Dr. Paulo Lício de Geus
UNICAMP – Campinas

Prof. Dr. Cesar Augusto Cavalheiro Marcondes
UFSCAR – São Carlos

São José do Rio Preto
27 de abril de 2016

Dedico este trabalho

Aos meus pais, Paulo e Rita, e a minha esposa Bárbara pelo incentivo, compreensão e amor durante todos os tempos.

AGRADECIMENTOS

Primeiramente agradeço a Deus por sua graça, pela vida e pelas condições que me trouxeram até aqui.

Agradeço a minha família por todo o suporte prestado, valores ensinados ao longo dos anos e pela paciência nos momentos difíceis. Agradeço também à minha amável esposa Bárbara por todo amor, carinho e por estar comigo em todas as situações, boas ou ruins.

Ao meu orientador Prof. Dr. Adriano Mauro Cansian pelos conhecimentos passados e pela orientação, contribuindo significativamente para minha formação acadêmica e pessoal.

Aos meus amigos e companheiros de laboratório Vinícius Vassoler Galhardi e Raphael Campos por todas as discussões e ensinamentos que tornaram este projeto possível.

Aos demais membros do laboratório: Amanda Barbosa, Bruno Leal, Leandro Gonçalves, Matheus Carreira, Pedro Ferracini e Rafael Carreira pelo companheirismo, pelas horas de conversa e estudo.

Agradeço ao Pr. José Genivaldo pela amizade e todos os seus conselhos e também aos amigos Daniel, William, Henrique, Adailton e Wellington e a todos os outros da família Missão Atos pelo companheirismo, as muitas risadas e os excelentes momentos vividos juntos.

Ao PPGCC – Programa de Pós-Graduação em Ciência da Computação e a todos os docentes do Departamento de Ciências de Computação e Estatística (DCCE) pelas disciplinas e conhecimentos transmitidos durante minha formação.

Agradeço também à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela bolsa de mestrado concedida para realização deste projeto.

“*Seja amável. Lembre-se de que todos os que você encontra estão enfrentando uma batalha difícil*”
Harry Thompson

RESUMO

Os Sistemas de Detecção de Intrusão baseados em rede (NIDS) são tradicionalmente divididos em dois tipos de acordo com os métodos de detecção que empregam, a saber: (i) detecção por abuso e (ii) detecção por anomalia. Aqueles que funcionam a partir da detecção de anomalias têm como principal vantagem a capacidade de detectar novos ataques, no entanto, é possível elencar algumas dificuldades com o uso desta metodologia. Na detecção por anomalia, a análise das anomalias detectadas pode se tornar dispendiosa, uma vez que estas geralmente não apresentam informações claras sobre os eventos maliciosos que representam; ainda, NIDSs que se utilizam desta metodologia sofrem com a detecção de altas taxas de falsos positivos. Neste contexto, este trabalho apresenta um modelo para a classificação automatizada das anomalias detectadas por um NIDS. O principal objetivo é a classificação das anomalias detectadas em classes conhecidas de ataques. Com essa classificação pretende-se, além da clara identificação das anomalias, a identificação dos falsos positivos detectados erroneamente pelos NIDSs. Portanto, ao abordar os principais problemas envolvendo a detecção por anomalias, espera-se equipar os analistas de segurança com melhores recursos para suas análises.

Palavras-chave: Classificação de Anomalias. Métodos de Agrupamento. Sistemas de Detecção de Intrusão baseados em Rede. Redução de Falsos Positivos.

ABSTRACT

Network Intrusion Detection Systems (NIDS) are traditionally divided into two types according to the detection methods they employ, namely (i) misuse detection and (ii) anomaly detection. The main advantage in anomaly detection is its ability to detect new attacks. However, this methodology has some downsides. In anomaly detection, the analysis of the detected anomalies is expensive, since they often have no clear information about the malicious events they represent; also, it suffers with high amounts of false positives detected. In this context, this work presents a model for automated classification of anomalies detected by an anomaly based NIDS. Our main goal is the classification of the detected anomalies in well-known classes of attacks. By these means, we intend the clear identification of anomalies as well as the identification of false positives erroneously detected by NIDSs. Therefore, by addressing the key issues surrounding anomaly based detection, our main goal is to equip security analysts with best resources for their analyses.

Keywords: Anomalies Classification. Clustering Methods. Network Intrusion Detection Systems. False Positives Reduction.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AC: *Anomalies Classifier*

AD: *Anomalies Detector*

AnoID: *Anomalies Identifier*

CA: Centrada em Ataques

CA: *Clustering Algorithm*

CFA: Centrada em Falsas Anomalias

CR: *Class Reader*

CSRF: *Cross-site request forgery*

DDoS: *Distributed Denial of Service*

DoS: *Denial of Service*

EM: *Expectation Maximization*

FA: Falsas Anomalias

FE: *Features Extractor*

FN: Falso Negativo

FP: Falso Positivo

HGP: Homogeneidade Global Ponderada

HIC: Homogeneidade Intra-Classe

IDMEF: *Intrusion Detection Message Exchange Format*

IDS: *Intrusion Detection System*

k-nn: *K-nearest neighbors*

KVM: *Kernel-based Virtual Machine*

NIDS: *Network Intrusion Detection System*

OPF: *Optimum-Path Forest*

OPFC: *Optimum-Path Forests Clustering*

PDF: *Probability Density Function*

ROC: *Receiver Operating Characteristic*

SLA: *Service Level Agreement*

SVM: *Support Vector Machines*

TVP: Taxa de Verdadeiros Positivos

VN: Verdadeiro Negativo

VP: Verdadeiro Positivo

CAPÍTULO 1 - Introdução

1.1 Considerações Iniciais

Atualmente, constata-se um vertiginoso crescimento no uso de serviços e aplicações executando sobre os sistemas computacionais, como consequência vê-se um aumento constante da quantidade de dados críticos que trafegam pelas redes de computadores, fazendo com que este se torne um valoroso e, portanto, visado meio de comunicação. Em decorrência disso vemos o aumento dos índices de incidentes de segurança, que se intensifica ano após ano como pode ser verificado através das estatísticas disponíveis em CERT.br - Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br, 2015), órgão do Comitê Gestor da Internet no Brasil. Em um cenário como este se justifica, cada vez mais, o emprego de mecanismos de defesa com o objetivo de detectar e possivelmente coibir ações maliciosas e ataques contra as redes de computadores.

Em meio aos mecanismos de defesa incluem-se os sistemas de detecção de intrusão baseados em rede (do inglês NIDS: *Network Intrusion Detection Sys*tems). Estes sistemas são responsáveis pelo monitoramento de um ambiente de rede e tem como objetivo detectar ações maliciosas sobre as informações ou serviços em execução neste ambiente. Os NIDS podem ser divididos em dois principais grupos de acordo com o tipo de técnica utilizada para a detecção: abuso e anomalia. Na detecção por abuso são geradas assinaturas que representam o comportamento de ataques já conhecidos.

Estas assinaturas são então confrontadas com o tráfego corrente na rede monitorada e sempre que o padrão representado por elas corresponde a qualquer atividade um ataque é detectado. Embora os NIDSs baseados em abuso sejam conhecidos por sua boa precisão de detecção, estes sistemas só detectam ataques para os quais existe uma assinatura, sendo esta sua principal desvantagem. Neste sentido, as metodologias baseadas em anomalia são mais eficientes. Uma vez que esta abordagem consiste no aprendizado do padrão normal do tráfego monitorado, é possível a detecção de qualquer desvio causado por um comportamento anômalo. Assim, ao contrário da detecção por abuso, os NIDSs baseados em anomalia têm a capacidade de detectar ataques desconhecidos para o sistema.

1.2 Motivação e Escopo

Apesar das vantagens citadas na seção anterior, os NIDSs baseados em anomalia sofrem por algumas dificuldades que são: 1) a detecção de uma elevada quantidade de falsos positivos (ELSHOUSH; OSMAN, 2010) e 2) a dificuldade em se analisar as anomalias detectadas, uma vez que estas geralmente não apresentam informações claras sobre os eventos maliciosos que representam (PAREDES-OLIVA et al, 2012).

Neste contexto, este trabalho apresenta um modelo para a classificação automática das anomalias que são detectadas por um NIDS em classes conhecidas de ataques. Com essa classificação têm-se dois principais objetivos: 1) a rápida identificação das anomalias detectadas no tráfego e 2) a redução do número de falsos positivos que são detectados pelo NIDS e encaminhados para análise, o que é feito pela classificação desses falsos alertas em meio a classes diferentes daquelas que representam ataques reais.

Para tal, fez-se o uso de algoritmos de aprendizagem de máquina não supervisionados, também conhecidos como métodos de agrupamento, ainda não utilizados para este fim. Esses métodos possuem uma grande capacidade de encontrar classes naturais em meio a um conjunto de dados e se

mostraram valiosos na separação das anomalias de acordo com suas categorias de ataque, além da separação entre os verdadeiros e falsos alertas.

Dessa forma, este trabalho inova ao ser o primeiro a realizar uma comparação entre diferentes métodos de agrupamento para fins de classificação de anomalias. É esperado que as comparações realizadas auxiliem futuros pesquisadores na escolha de seus algoritmos para suas pesquisas na área. Ademais, elaborou-se um modelo útil não só para a classificação das anomalias desconhecidas, mas também para a redução dos falsos positivos e o consequente aumento da precisão de detecção.

Finalmente, os resultados obtidos podem contribuir para uma análise mais rápida e eficiente (PAREDES-OLIVA et al, 2012), a tomada de contramedidas de forma automática (XING et al, 2013) e o uso de ferramentas de correlação para a construção de cenários de ataques (MIRSHAHJAFARI; GHAVAMNIA, 2014). Portanto, abordando os principais desafios da detecção de intrusão baseada em anomalia, esta pesquisa tem como objetivo equipar os analistas de segurança com melhores recursos para suas análises.

1.3 Organização da dissertação

Esta dissertação está dividida em seis capítulos, incluindo o atual. No Capítulo 2 é realizada uma fundamentação teórica a fim de contextualizar o leitor acerca de conceitos e tecnologias que foram importantes para o desenvolvimento deste trabalho. No Capítulo 3 são apresentados os trabalhos relacionados ao tema de classificação de anomalias e também algoritmos de agrupamento no contexto da classificação de tráfego legítimo. No Capítulo 4 são apresentados os objetivos e a metodologia utilizada por esta pesquisa. No Capítulo 5 são explicados os experimentos realizados e discutidos os resultados obtidos. Por fim, no Capítulo 6 são expostas as conclusões acerca do tema e dos resultados obtidos.

CAPÍTULO 2 - Fundamentação Teórica

2.1 Considerações iniciais

Este capítulo tem por objetivo apresentar a fundamentação teórica necessária para o entendimento dos tópicos presentes neste estudo e dos resultados apresentados. Na Seção 2.2 são apresentados alguns conceitos relacionados a ataques em redes de computadores, bem como uma descrição dos ataques utilizados nesta pesquisa. Na Seção 2.3 são abordados alguns conceitos importantes concernentes aos sistemas de detecção de intrusão. Na Seção 2.4 discute-se algumas métricas importantes na avaliação de sistemas de detecção e classificação. Na Seção 2.5 é discorrido sobre os métodos de agrupamento com ênfase nos algoritmos utilizados neste trabalho. Por fim, na Seção 2.6 são apresentados conceitos relacionados a fluxo de dados que foram utilizados para a coleta de dados nos testes realizados.

2.2 Ataques às redes de computadores

Ataques no contexto da segurança da informação podem ser definidos como quaisquer tentativas, tendo elas sucesso ou não, de subverter a autenticidade, a confidencialidade, a integridade e a disponibilidade de um sistema computacional (GOLLMAN, 1999). Este conceito é igualmente

aplicado para as redes de computadores, que são alvos de muitos tipos de ataques com os mais variados objetivos. Este trabalho dará destaque para cinco tipos de ataques bem difundidos hoje em dia e que foram utilizados para testes dos sistemas desenvolvidos.

2.2.1 Ataques de força bruta

Os ataques de força bruta (PINKAS; SANDER, 2002) estão associados a tentativas sucessivas e exaustivas de se tentar descobrir uma identificação válida para a autenticação de um sistema. Exemplo disso são os usuários e senhas de serviços como o SSH e o FTP, os quais são largamente utilizados na Internet. Essa classe de ataque pode ser dividida principalmente em duas categorias, que são:

- Ataque de força bruta: esse é o ataque em toda a sua essência. Ele consiste em testar todas as possibilidades de autenticação de um sistema. Este método pode ser extremamente ineficaz, considerando o atraso da comunicação nas redes, a força da senha, entre outros.
- Ataques de dicionário: é uma variante mais otimizada do ataque de força bruta. Seu funcionamento consiste na utilização de uma lista de palavras, previamente construída. A esta lista dá-se o nome de dicionário. Nesta categoria, os ataques são geralmente feitos com a fixação de um usuário e o teste de cada palavra do dicionário como possível senha válida. Pode-se, eventualmente, variar cada palavra do dicionário, adicionando números ou fazendo pequenas trocas de caracteres, o que pode aumentar as chances de sucesso do ataque. Este ataque é geralmente mais eficiente que o primeiro devido ao uso indiscriminado de senhas padrão em dispositivos diversos e a escolha de senhas fracas, geralmente associadas a nomes, datas e palavras comuns do cotidiano. No entanto, a eficácia deste ataque se limita a qualidade do dicionário utilizado.

2.2.2 Ataques de negação de serviço

Os ataques de negação de serviço (SCHUBA et al, 1997), mais conhecidos como DoS (do inglês DoS: *Denial of Service*), prezam pela subversão da disponibilidade dos sistemas atacados. No contexto das redes de computadores, este processo se dá por dois principais meios: 1) a diminuição da largura de banda ou 2) o esgotamento de recursos. Os servidores Web são alvos bastante comuns deste tipo de ataque, aos quais são enviados uma grande quantidade de solicitações de conexão até o ponto que sejam incapazes de responder por novas solicitações, mesmo aquelas legítimas. Esses ataques são geralmente executados de forma distribuída, o que potencializa suas chances de sucesso, caracterizando um ataque distribuído de negação de serviço ou DDoS (do inglês DDoS: *Distributed Denial of Service*) (MIRKOVIC; REIHER, 2004).

2.2.3 Ataques de varredura

Muito se discute se os eventos de varredura em redes se caracterizam como ataque, uma vez que em tese não apresentam comportamento intrusivo, pois consistem basicamente num levantamento prévio de informações. No entanto, é comum que os eventos de varredura precedam outros ataques, pois as informações obtidas permitem um planejamento mais efetivo de ataques posteriores, geralmente de maior risco. Por este motivo, esses eventos serão referenciados como ataques no restante deste trabalho.

Nos ataques de varredura (LEE; ROEDEL; SILENOK, 2003), o atacante visa o levantamento acerca dos dispositivos computacionais ativos na rede, bem como os serviços por eles executados. Estes ataques podem se utilizar de técnicas simples, como o envio de mensagens *icmp* a endereços pseudoaleatórios a fim de se descobrir os *hosts* comunicantes em uma certa rede, além de tentativas de conexão a um certo range de portas de um *host* específico com o objetivo de se descobrir os serviços por ele executados. No entanto, também pode-se utilizar de técnicas mais furtivas, como varreduras discretas (do inglês *Low-Profiling Probing*) (TREURNIET, 2006), com o propósito de se dificultar a detecção de análise destas.

2.2.4 Varredura de vulnerabilidades Web

Estes ataques são geralmente realizados com o uso de ferramentas automáticas para a exploração de vulnerabilidades em sistemas Web. Essas ferramentas (FONSECA et al, 2007) são geralmente utilizadas pelos desenvolvedores em testes de suas aplicações *Web*. O seu uso de forma maliciosa contra sistemas de terceiros pode revelar vulnerabilidades passíveis de serem exploradas por ataques posteriores.

2.3 Sistemas de detecção de intrusão baseados em rede

Os sistemas de detecção de intrusão baseados em redes pertencem a uma subcategoria dos tradicionais sistemas de detecção de intrusão (do inglês IDS: *Intrusion Detection System*). Um IDS tem como função o monitoramento de um ambiente que seja alvo potencial de um ataque. Neste sentido, os NIDSs são os IDSs responsáveis pelo monitoramento de ambientes de rede. Dessa forma, os conceitos concernentes aos IDSs são automaticamente herdados pelos NIDSs.

Além da classificação quanto ao ambiente de monitoramento, a classificação mais tradicional dos IDSs é quanto às suas metodologias de detecção, podendo ser elas baseadas em abuso ou anomalia. Os autores em (ELSHOUSH; OSMAN, 2010) fazem uma importante análise comparativa entre esses dois métodos.

A detecção por abuso se fundamenta no conceito de *Pattern-Matching*. Nesta técnica, busca-se a geração dos padrões das intrusões que se queira detectar por meio de assinaturas. Essas assinaturas são então confrontadas com as atividades correntes do sistema: caso a intrusão previamente modelada ocorra e combine com a assinatura gerada a detecção será realizada. Esta abordagem tem como vantagem a alta precisão alcançada devido ao número reduzido de falsos positivos que são detectados. Como

principal desvantagem, destaca-se a detecção somente das intrusões com os padrões previamente conhecidos.

Por outro lado, na detecção baseada em anomalia a meta é a procura por desvios em algumas medidas estatísticas com o objetivo de se detectar comportamentos não usuais. Em uma etapa inicial devem ser colhidos alguns dados capazes de definir certos tipos de comportamentos do ambiente em análise. Esses dados são então submetidos a uma etapa conhecida como “fase de treinamento”. Nesta fase são estabelecidos os limites do comportamento normal e então definidos os limiares para a detecção dos eventos. Como a detecção baseada em anomalias consiste na detecção de tudo aquilo que foge ao comportamento considerado normal, esta técnica possui a capacidade de detectar ataques não previamente conhecidos, o que constitui sua principal vantagem. No entanto, devido às dificuldades de se determinar os limiares daquilo que pode ser considerado legítimo (CHANDOLA; BANERJEE; KUMAR, 2009), esta técnica costuma apresentar altos índices de falsos positivos. Além disso, a análise dos eventos detectados costuma ser bastante dispendiosa (PAREDES-OLIVA et al, 2012). Outras classificações concernentes aos IDSs podem ser encontradas em (WU; BANZHAF, 2010).

O desempenho dos sistemas de detecção é geralmente atestado por medidas clássicas dos sistemas de classificação, uma vez que sua eficácia se mede pela sua capacidade de realizar boas predições (WU; BANZHAF, 2010). Estas métricas são discutidas na seção a seguir.

2.4 Métricas para avaliação de desempenho

A fim de se avaliar o desempenho do modelo proposto, foram empregadas algumas métricas clássicas na literatura, em especial aquelas utilizadas na avaliação de sistemas de detecção de intrusão (WU; BANZHAF, 2010).

Em um problema binário, que é bastante típico da detecção de anomalias, sempre que um caso positivo é corretamente classificado têm-se

um caso de verdadeiro positivo (VP); quando não, têm-se um caso de falso negativo (FN). Já com relação aos casos negativos, quando uma classificação é correta têm-se um caso de verdadeiro negativo (VN); quando não é, verifica-se um caso de falso positivo (FP).

Não obstante, este trabalho lida com um problema multiclasse, onde as anomalias devem ser classificadas de acordo com seu tipo de ataque e não somente em termos de comportamento anômalo e não anômalo. Assim sendo, os casos de VPs, FNs, VNs e FPs devem ser contabilizados para cada uma das diferentes classes do modelo de classificação. Ou seja, o número de VPs de uma classe x (i.e. VP(x)) é igual ao número de membros da classe x corretamente classificados como pertencentes à classe x. FN(x) é igual ao número de membros da classe x incorretamente classificados como não pertencentes a x. VN(x) é o número de membros de outras classes corretamente classificados como não pertencentes a x. FP(x) é o número de instâncias de outras classes incorretamente classificados como pertencentes a X.

Os casos descritos acima também podem ser visualizados em uma matriz especial, que é conhecida como matriz de confusão (ELSHOUSH; OSMAN, 2010) e que pode ser vista na Tabela 2-1.

Tabela 2-1. Matriz de Confusão.

	Valores Preditos	
Valores Reais	Casos Negativos (Normais)	Casos Positivos (Ataques)
Casos Negativos (Normais)	**VN**	**FP**
Casos Positivos (Ataques)	**FN**	**VP**

Com os valores da matriz de confusão é possível a criação de métricas bastante significativas:

- Acurácia: considera todos os casos corretamente classificados sobre o total de casos, sendo utilizada para mensurar a qualidade geral de um modelo. Num contexto multiclasse, define-se a Acurácia Global, que considera o total de acertos sobre o total de casos em todas as classes do conjunto *C* de um modelo de classificação.

$$Acurácia = \frac{VN + VP}{VN + VP + FN + FP} \quad ou \tag{2.1}$$

$$Acurácia\ Global = \frac{\sum_c VP_c}{\sum_c VP_c + \sum_c FP_c}.$$

- Sensibilidade ou Taxa de Verdadeiros positivos (TVP): considera todos os casos positivos e corretamente classificados sobre o total de casos positivos. Num problema multiclasse esta métrica deve ser usada para a avaliação de cada classe encontrada.

$$TVP = \frac{VP}{VP + FN} \quad ou \quad TVP(x) = \frac{VP(x)}{VP(x) + FN(x)}. \tag{2.2}$$

- Precisão: considera todos os casos positivos e corretamente classificados sobre o total de casos classificados como positivos. Em um problema multiclasse esta métrica também deve ser usada para mensurar cada classe do modelo.

$$Precisão = \frac{VP}{VP + FP} \quad ou \quad Precisão(x) = \frac{VP(x)}{VP(x) + FP(x)}. \tag{2.3}$$

- Taxa de Falsos Positivos (TFP): Considera todos os falsos positivos sobre o total de casos negativos.

$$TFP = \frac{FP}{VN + FP}. \tag{2.4}$$

- Medida-F (*F-Measure*): é o resultado da média harmônica entre a *TVP* e *Precisão*. Pode-se usar essa métrica para calcular o equilíbrio entre estas duas medidas, uma vez que muitas vezes se verifica uma relação inversa entre elas.

$$Medida - F = \frac{2 * Precisão * TVP}{Precisão + TVP}. \quad (2.5)$$

- Homogeneidade Intra-Classe (HIC): esta métrica foi definida pelos autores em (ZANDER et al, 2005) e considera os elementos da classe mais frequente em um certo grupo sobre o número total de elementos deste grupo. Isto é, sejam *A* e *C* conjuntos das aplicações classificadas e das classes geradas no treinamento, respectivamente. Define-se também a função $count(a, c)$ que conta o número de casos da aplicação $a \in A$ classificados na classe $c \in C$. Então a homogeneidade intra-classe $H(c)$ de uma classe c é definida como a maior fração de casos de uma aplicação na classe, o que é representado pela Equação 2.6:

$$H(c) = \frac{\max(count(a, c) | a \in A)}{\sum_a count(a, c)}. \quad (2.6)$$

- Homogeneidade Global Ponderada (HGP): esta métrica foi desenvolvida especialmente para esta pesquisa. A HGP foi elaborada para a avaliação da qualidade dos modelos de classificação obtidos pelos algoritmos na fase de treinamento. Esta métrica computa a média ponderada de HIC correspondente a um conjunto de classes. Observou-se que a simples média de HIC era fortemente influenciada por grupos com pouco número de instâncias. Como estes grupos tendem a representar somente casos bastante específicos, verificou-se a pouca influência destes na etapa de classificação. Dessa forma, optou-se por ponderar a

HIC de cada grupo de acordo com a sua relevância para a fase de classificação, que é medida pelo total de elementos classificados em cada grupo durante a etapa de treinamento. Para isso, considere $\max\_instances(C)$ a função que retorna o número máximo de instâncias agrupadas por uma das classes do conjunto C. Então, o peso $w(c)$ para cada classe $c \in C$ pode ser computado como:

$$w(c) = \frac{\sum_a count(a, c)}{\max\_instances(C)}. \quad (2.7)$$

Assim, a HGP para um conjunto de classes, encontradas na etapa de treinamento de um algoritmo de agrupamento pode ser definida da seguinte forma:

$$HGP(C) = \frac{\sum_c H(c)w(c)}{\sum_c w(c)}. \quad (2.8)$$

2.5 Algoritmos de agrupamento

Os algoritmos de agrupamento (do inglês *Clustering Algorithms*), também conhecidos como algoritmos de aprendizagem de máquina não-supervisionados (GENTLEMAN; CAREY, 2008), são técnicas de mineração de dados que visam a classificação e o agrupamento de dados considerando suas características similares. O principal objetivo destes métodos é a divisão dos dados em grupos, também conhecidos como classes. Cada classe compreende objetos que são similares entre si e diferentes dos objetos de outras classes.

Os métodos de agrupamento possuem a grande habilidade de identificar as classes naturais de um certo problema (ZANDER et al, 2005), o que os difere dos métodos supervisionados, onde as classes precisam ser determinadas anteriormente à etapa inicial de aprendizagem. Esta

capacidade foi explorada por este trabalho para a classificação de anomalias em tráfego de rede, pois de acordo com os trabalhos [(ZANDER et al, 2005), (NGUYEN; ARMITAGE, 2008)] o problema de classificação envolvendo tráfego de rede implica a geração de várias classes por aplicação. Assim optou-se pelos métodos de agrupamento para que estes possam encontrar um melhor modelo de classificação, utilizando-se de quantas classes forem necessárias para a obtenção de melhores resultados. Os métodos elencados para uso nesta pesquisa são explicados nas seções seguintes.

2.5.1 OPF não-supervisionado

O classificador OPF (do inglês OPF: *Optimum-Path Forests*) consiste num método baseado em grafos que explora as relações de conectividade entre um conjunto de amostras em um certo espaço de características. Sua metodologia considera um conjunto de treinamento como um grafo, sendo que os nós são as amostras deste conjunto e os arcos são definidos de acordo com alguma relação de adjacência. Este grafo é então particionado em árvores de caminhos ótimos (do inglês OPT*: Optimum Path Tree*) com raízes nos nós mais representativos, denominados protótipos. Para a definição das OPTs os protótipos competem entre si pelos nós mais fortemente conectados, sendo que uma instância é conquistada pelo protótipo que lhe oferecer um caminho ótimo, ou seja, de menor custo de acordo com uma certa função de conectividade. Todos os nós pertencentes a uma OPT herdam o rótulo de seu respectivo protótipo. Dessa forma, as OPTs representam as diferentes classes no problema de classificação e o seu conjunto é responsável pelo nome do classificador OPF. No OPF, a classificação de uma nova instância consiste basicamente em encontrar o protótipo, definido pela fase de treinamento, que lhe ofereça o melhor caminho dentre todos os outros oferecidos.

A metodologia OPF oferece um método supervisionado (PAPA et al, 2009) e um não supervisionado (ROCHA et al, 2009), e tem sido validada com sucesso em diferentes campos. Seguindo a proposta inicial, este trabalho se utilizou da versão não supervisionada, também conhecida como OPFC (do

inglês OPFC: *Optimum-Path Forests Clustering*), cuja teoria pode ser consultada a seguir.

Considere Z um conjunto de instâncias tal que para toda instância $s \in Z$ exista um vetor de características $\vec{v}(s)$. Seja $d(s,t)$ a distância entre as instâncias *s* e *t* no espaço de características. Por padrão o OPFC considera $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$ como sendo a distância Euclidiana entre $\vec{v}(t)$ e $\vec{v}(s)$, tal escolha foi mantida por este trabalho, no entanto, outras distâncias podem ser escolhidas para sua utilização.

Com as definições acima, pode-se estabelecer um grafo (Z, A_k), de tal modo que duas amostras *s* e *t* são adjacentes (i.e. $(s,t) \in A_k$), quando satisfazem a seguinte relação de adjacência A_k:

$$\boldsymbol{t \in A_k(s)\ se\ t\ é\ k\ vizinho\ mais\ próximo\ de\ s\ no\ espaço\ de\ características.}$$

Deste modo, é definido um grafo do tipo k-nn (do inglês k-nn: *K-nearest neighbors*) (DONG et al, 2011) de parâmetro *k*, o qual precisa ser encontrado de acordo com cada tipo de aplicação. Para encontrar o melhor valor para *k* (i.e. k^*), Rocha et al. (ROCHA et al, 2009) propõem um método que considera o corte mínimo do grafo entre todos os resultados para $k^* \in [1, k_{max}]$ de acordo com a medida normalizada sugerida em (SHI; MALIK, 2000). Ainda assim, a escolha de k fica condicionada ao parâmetro k_{max} que deve ser fornecido pelo usuário. A escolha de k_{max} para a utilização do OPFC nesta pesquisa é detalhada na Seção 5.4.

No grafo (Z, A_k) os arcos são ponderados por $d(s,t)$ e os nós $s \in Z$ são ponderados pelos valores resultantes da função densidade de probabilidade (do inglês PDF: *Probability Density Function*) $\rho(s)$, definida pela Equação 2.9:

$$\rho(s) = \frac{1}{\sqrt{2\pi\sigma^2}|A_k(s)|} \sum_{t \in A(s)} \exp\left(\frac{-d^2(s,t)}{2\sigma^2}\right), \quad (2.9)$$

onde $|A_k(s)| = k$, $\sigma = \frac{d_f}{3}$, sendo d_f o arco de maior peso em (Z, A_k).

Para a escolha dos protótipos, o OPFC seleciona as amostras com valores máximos da PDF ρ. Uma vez que um certo máximo da PDF pode ser um subconjunto de amostras, formando-se assim um platô, e sabendo-se que a relação A_k é assimétrica, adiciona-se arcos simétricos nos platôs para se garantir um único protótipo por máximo da PDF.

Para a geração das OPTs, o OPFC define um caminho π_t como uma sequência de amostras adjacentes com início em $R(t)$ – conjunto de amostras raízes (i.e. protótipos) – e término em uma instância t. O caminho $\pi_t = < t >$ é dito trivial e $\pi_s . < s, t >$ é a concatenação de π_s e o arco (s, t). A função de conectividade $f(\pi_t)$ atribui a todo π_t um valor que representa a força de conexão entre os nós do caminho. Um caminho π_t é considerado ótimo quando $f(\pi_t) \geq f(\tau_t)$ para qualquer outro caminho τ.

O OPFC então atribui a cada instância t o caminho π_t cujo valor de densidade mínimo é o máximo dentre todos os oferecidos. Ou seja, o método busca encontrar $V(t) = max_{\forall \pi_t \in (Z, A_k)}\{f(\pi_t)\}$, sendo $f(\pi_t)$ definida pela Equação 2.11:

$$f(< t >) = \begin{cases} \rho(t) & se\ t \in R \\ \rho(t) - \delta & caso\ contrário, \end{cases} \qquad (2.10)$$

$$f(< \pi_s . < s, t >>) = \min\{f(\pi_s), \rho(t)\}.$$

O parâmetro δ tem como objetivo reduzir a influência dos domos da PDF com altura menor que δ, com isso as amostras nestes domos não seriam eleitas protótipos e o resultado final seria, teoricamente, a remoção de classes irrelevantes. No entanto, alguns testes mostraram que o problema da classificação de anomalias é bastante sensível a este parâmetro, sendo que valores $\delta > 0$ implicaram na eliminação de classes importantes, assim este trabalho considerou $\delta = 0$ para todos os testes.

Por fim, para todas as amostras $t \in Z$ existirá um caminho ótimo $P^*(t)$ que é trivial, se $t \in R$, ou possui a forma $P^*(s) . < s, t >$ onde:

a) $f\big(P^*(s)\big) \geq f\big(P^*(t)\big)$,

b) $P^*(s)$ é ótimo,

c) Para qualquer caminho ótimo $P^*(s), f(P^*(s).< s,t >) = f\big(P^*(t)\big) = V(t)$.

Dessa forma, os caminhos ótimos P representam as OPTs e cada OPT representa uma classe a ser utilizada pelo processo de classificação pelo OPFC. Como implementação para o OPFC se utilizou a versão disponível na LibOPF (LibOPF, 2016).

2.5.2 Algoritmo AutoClass

O AutoClass é um algoritmo de agrupamento particional e classificação *fuzzy* que se baseia no modelo bayesiano para criação de seu modelo de classificação. Em sua fase de treinamento, o AutoClass aprende de forma automática as classes naturais inerentes a um conjunto de dados. Este processo se dá pelo agrupamento das instâncias com características semelhantes e tem como resultado um conjunto de classes que podem ser usadas posteriormente para classificação de novos dados. Nesta seção é providenciada uma visão geral sobre o Autoclass, para maiores detalhes o leitor é encorajado a consultar (CHEESEMAN; STUTZ, 1996).

O método bayesiano para classificação não supervisionada estabelece um modelo probabilístico no qual as instâncias são atribuídas a cada classe de acordo com um grau de pertinência expresso na forma de uma probabilidade. Seja $X = \{X_1, \dots, X_i\}$ um conjunto de instâncias, onde cada instância X_i pode ser representada por um vetor ordenado de características $\overrightarrow{X_\iota} = \{X_{i1}, \dots, X_{ik}\}$. Na classificação não-supervisionada o desafio é encontrar o mais provável modelo H que descreva conjunto X, ou seja, o problema consiste em encontrar H tal que $P(H \mid X)$ seja máximo. No modelo bayesiano, a probabilidade $P(H \mid X)$ é conhecida como probabilidade posterior de H e é proporcional a probabilidade $P(X \mid H)$, conhecida como função de verossimilhança *(likelihood function)*. O teorema de Bayes é expressado de acordo com a Equação 2.12:

$$p(H \mid X) = \frac{p(H)\, p(X \mid H)}{p(X)}. \qquad (2.11)$$

As probabilidades $p(H)\ e\ p(X)$ são denominadas probabilidades a priori e consistem em probabilidades iniciais, antes da observação de novas evidências, como por exemplo o modelo *H* e o conjunto *X*.

No contexto do AutoClass, o modelo *H* representa a quantidade e os descritores das classes cotadas para serem as mais descritivas para a representação do conjunto X. Dessa forma, o algoritmo busca pelo *H* que maximiza a probabilidade a posteriori $p(H \mid X)$.

Para compor a probabilidade final de que uma instância X_i pertença a uma certa classe C_j, de um conjunto com J classes, o AutoClass se utiliza de duas probabilidades: a probabilidade interclasse e a probabilidade intraclasse. Como o conjunto J consiste num espaço particionado de dados, a função de probabilidade interclasse é definida por uma distribuição de Bernoulli, caracterizada por um certo conjunto $\overrightarrow{V_c}$ de probabilidades $\{\pi_1, \dots, \pi_c\}$, dado que $0 \leq \pi_j \leq 1$ e $\sum_j \pi_j = 1$. Assim a função de probabilidade intraclasse é definida de acordo com a Equação 2.13:

$$p\left(X_i \in C_j \mid \overrightarrow{V_c}\right) \equiv \pi_j. \qquad (2.12)$$

A probabilidade intraclasse consiste no produto de distribuições de probabilidade das k características de cada vetor $\overrightarrow{x_i}$. O AutoClass realiza a forte suposição de que as características são condicionalmente independentes, embora esta seja uma característica difícil de se obter no mundo real, os bons resultados já demonstrados pelo AutoClass o tornam apto para esta pesquisa. Com esta suposição a probabilidade intraclasse pode ser representada de acordo com a Equação 2.14:

$$P\left(\overrightarrow{X_i} \mid X_i \in C_j, \overrightarrow{V_j}\right) = \prod_k P(X_{ik} | X_i \in C_j, \overrightarrow{V_{jk}}). \qquad (2.13)$$

Nesta pesquisa, as características usadas para a classificação são valores reais, assim, foram modeladas com distribuições log-normal. Dessa forma, a

probabilidade final de que uma instância X_i com características $\overrightarrow{X_i}$ pertença a uma dada classe C_j é dada pela Equação 2.15:

$$P(\overrightarrow{X_i}, X_i \in C_j \mid \overrightarrow{V_c}, \overrightarrow{V_j}) = \pi_j \prod_k P(X_{ik} \mid X_i \in C_j, \overrightarrow{V_{jk}}). \quad (2.14)$$

Para a resolução da probabilidade expressa pela Equação 2.15 o AutoClass precisa basicamente encontrar os valores do vetor de parâmetros $\vec{V}$ e a quantidade de classes *J* que melhor representa os dados do conjunto X. Para tal, o AutoClass busca pela máxima probabilidade a posteriori de $\vec{V}$, definida pela Equação 2.16:

$$P(\vec{V} \mid X) = \frac{P(X, \vec{V})}{P(X)} = \frac{P(X, \vec{V})}{\int d\vec{V}\, P(X, \vec{V})}. \quad (2.15)$$

Isso por se considerar o teorema da probabilidade total, que afirma que se os eventos em $\vec{V}$ são mutuamente exclusivos, então $P(X) = \int d\vec{V}\, P(X, \vec{V})$, considerando que o espaço $\vec{V}$ é contínuo.

O AutoClass pode ser configurado com um número pré-definido de classes ou pode estimar este número automaticamente, sendo esta uma de suas grandes características. Para o problema da classificação de anomalias e redução de falsos positivos, configurou-se o AutoClass para a busca automática das classes, uma vez que o tráfego considerado legítimo pode ser bem heterogêneo e diverso, o que torna impraticável a definição manual das classes.

Quando o número de classes J é desconhecido, o AutoClass deve ser configurado com uma lista inicial J_{lista} com possíveis valores para J com os quais o algoritmo realizará as primeiras buscas pelos parâmetros $\vec{V}$. Para cada possível J, o AutoClass utiliza uma variação do algoritmo de Maximização da Esperança (do inglês EM: *Expectation Maximization*) proposto por Dempster et al (DEMPSTER et al, 1977). Em sua busca, o algoritmo EM converge para um dos máximos locais da função $P(\vec{V} \mid X)$. Como pode haver vários desses

pontos, o AutoClass gera pontos pseudoaleatórios no espaço de parâmetros e inicia um novo ciclo com o Algoritmo EM. Esta operação se repete até que se convirja para o máximo global da função, o que leva em torno de 10-100 ciclos.

Ao término dos elementos de J_{lista}, gera-se uma distribuição log-normal com as dez melhores classificações até o momento. A partir desta distribuição o AutoClass passa a escolher randomicamente os novos possíveis valores de *J* para os próximos testes. O AutoClass encerra suas buscas quando o número total de ciclos, somado entre todas as iterações do algoritmo, alcança o valor estabelecido pelo parâmetro *max_n_tries*, ou o tempo total de execução alcança o tempo determinado pelo parâmetro *max_duration*; ambos os parâmetros podem ser definidos pelo usuário. Para este trabalho, definiu-se a quantidade de oitocentos ciclos como critério de parada. Ao término o Autoclass determina o vetor $\vec{V}$ que maximiza a função $P(\vec{V} \mid X)$, definindo assim seu modelo de classificação. Como implementação para este algoritmo se utilizou a última versão oficial, disponibilizada em (AutoClass C, 2016).

2.5.3 Algoritmo K-médias

O K-médias é uma técnica clássica na literatura cujo método de aprendizagem organiza um conjunto de objetos em um conjunto fixo de *K* partições. Cada partição define um grupo no problema de agrupamento. Este algoritmo utiliza uma abordagem bastante simples e, por conseguinte, é uma das abordagens mais rápidas.

Para o particionamento dos dados, considere $X = \{x_1, \dots, x_i\}$ um conjunto n-dimensional com *i* instâncias, modeladas como pontos, a serem particionadas entre um conjunto $C = \{c_1, \dots, c_k\}$ com *K* centroides. Cada ponto $x \in X$ é atribuído ao centroide mais próximo, e cada coleção de pontos atribuído a um centroide é um grupo ou aglomerado.

Para o encontro do melhor modelo de aglomerados, considere μ_k a média ou centroide de todos os pontos no grupo c_k. O K-médias tem então como objetivo a minimização da soma do erro quadrático entre μ_k e os pontos

no aglomerado c_k para todos os *K* grupos. A soma $J(C)$ do erro quadrático, num conjunto C, é definida pela Equação 2.17:

$$J(C) = \sum_{k=1}^{K} \sum_{x_i \in c_k} \|x_i - \mu_k\|^2. \quad (2.16)$$

Para isso o K-médias estabelece um conjunto inicial de centroides de forma pseudoaleatória a partir de uma semente. A partir deste conjunto são realizadas iterações, atualizando-se a posição de cada centroide para o seu ponto μ_k. Isso ocorre até que se verifique o mesmo conjunto de pontos atribuídos a cada centroide em duas rodadas consecutivas. Um simples algoritmo deste processo é descrito abaixo:

1. Inicialize K pontos como centroides no conjunto de pontos.
2. Atribua as instâncias ao centroide mais próximo de acordo com a função de distância Euclidiana.
3. Recalcule os centroides a partir da média dos pontos de cada grupo formado.
4. Repita os passos 2 e 3 até que não haja mais mudanças na atribuição dos pontos aos centroides.

Para operar, o K-médias requer a especificação de basicamente dois parâmetros, o número total de aglomerados *K* e um valor semente para o estabelecimento do conjunto inicial de centroides. A definição dos valores K utilizados neste projeto é descrita juntamente com os experimentos apresentados no Capítulo 5. Com relação ao valor de semente, notou-se a convergência ao máximo global, independentemente do valor escolhido. Com relação a implementação, foi utilizada a versão Java do K-médias disponibilizada pela suíte WEKA (WITTEN; FRANK, 2000).

2.5.4 Algoritmo X-médias

O algoritmo X-médias consiste numa extensão do algoritmo K-médias que implementa um meio automático para a determinação do parâmetro K.

Para operar, este algoritmo requer somente um intervalo $[K_{min}, K_{max}]$ com possíveis valores para *K*.

Para a construção de seu particionamento, o algoritmo desempenha as seguintes operações até a conclusão:

1. Melhoria de parâmetros (*Improve-Params*).
2. Melhoria de estrutura (*Improve-Structure*).
3. Se $K > K_{max}$ termine a operação e relate o melhor modelo obtido.

A etapa de melhoria de parâmetros consiste na execução do K-médias tradicional, já a melhoria de estrutura apura a necessidade de se adicionar novos centroides ao modelo obtido pelo K-médias na melhoria de parâmetros. Para tanto, a melhoria de estrutura é iniciada com a divisão de cada centroide em dois centroides filhos, eles são deslocados no sentido oposto de uma direção escolhida aleatoriamente. A distância deste deslocamento é proporcional ao tamanho da região do aglomerado definido pelo centroide pai. Após, é aplicada uma execução local do K-médias em cada região, com *K* igual a dois, para cada par de filhos. O algoritmo então realiza uma avaliação em todos os pares de filhos com o objetivo de se verificar o benefício ou não da divisão previamente realizada. De acordo com os resultados verificados, o centroide pai, ou seus filhos, são eliminados. Ao término desta etapa, volta-se para a melhoria de parâmetros. Este processo se desenvolve até que seja alcançado o valor de K_{max}, quando então o algoritmo termina e é retornada a melhor partição avaliada.

As partições obtidas em cada uma das iterações são avaliadas de acordo com o índice BIC (MIRKIN, 2005), que é aplicado de forma global às partições obtidas e também localmente, nas partições obtidas pelas divisões do centroide. Neste trabalho utilizou-se a implementação Java do X-médias também disponibilizada pela suite WEKA.

2.6 Fluxo de dados

Uma iniciativa que tem mostrado grande eficácia no monitoramento de redes de grande porte é a análise de informações através da exportação de fluxos de dados da rede. Esta é uma tecnologia que sumariza e contabiliza características em comum contidas nos pacotes. Com isso é disponibilizado um meio eficiente e escalável para análises de rede, com a vantagem de não infringir a privacidade do usuário ao não se utilizar do campo de dados dos pacotes.

A padronização dos campos que definem um fluxo foi criada pelo IETF (do inglês IETF: *Internet Engineering Task Force – Internet Society*), através do protocolo IPFIX (CLAISE, 2008). Baseado neste padrão, destaca-se o *Netflow* (CLAISE, 2004), criado pela *Cisco Systems*. O *Netflow* define um fluxo como sendo uma sequência unidirecional de pacotes entre dois pontos de comunicação. Desta padronização também emergiu a tecnologia de fluxos bidirecionais, representada pelo *Biflow* (TRAMMELL, 2008), em que um fluxo representa pacotes fluindo em ambas as direções de uma conexão de rede.

O posicionamento estratégico, em uma rede de computadores, de sensores, denominados exportadores de fluxo, possibilita a sumarização de características comuns presentes nos pacotes: informações que enviadas a um coletor (CORRÊA; PROTO; CANSIAN,2008) formam uma rica base para mineração de dados, análise de tráfego, detecção de eventos maliciosos, entre outros.

2.7 Considerações Finais

Nesta seção foram apresentados conceitos importantes para o bom entendimento desta pesquisa. A forma como estas tecnologias e ideias foram aplicadas é descrita no decorrer dos próximos capítulos.

CAPÍTULO 3 - Trabalhos Relacionados

3.1 Considerações iniciais

Neste capítulo são apresentados os trabalhos relacionados com o trabalho apresentado nesta dissertação. São discutidos trabalhos tanto da área de classificação de anomalias como da área de classificação de tráfego legítimo. Pelo fato de um ataque sobre uma rede se caracterizar como um tráfego de rede com suas características próprias, as áreas de classificação de anomalias e tráfego legítimo de rede são bastante correlatas.

3.2 Classificação automática de anomalias

A detecção baseada em anomalias tem sido amplamente estudada pela comunidade científica. Isso pode ser visto pela extensa revisão feita pelos autores em (AHMED et al, 2016) e também por recentes trabalhos, como os publicados em [(LIN et al, 2014); (ZHANG et al, 2015), (JEONG et al, 2016)]. No entanto, a área de classificação de anomalias de tráfego é ainda bastante inexplorada, com poucos trabalhos encontrados na literatura (PAREDES-OLIVA et al, 2012).

Lakhina et al. (LAKHINA et al, 2005) foram pioneiros ao mostrar a factibilidade da classificação de anomalias por meio de métodos automáticos.

Neste trabalho, os autores mostraram que diferentes anomalias podem ser categorizadas em classes diferentes por meio de algoritmos de agrupamento. As diferenças entre as anomalias são refletidas em suas características de tráfego. Em tal trabalho, estas diferenças foram percebidas pelos algoritmos *K-Médias* e *Hierarchical Agglomeration,* que realizaram uma mineração dos padrões de anomalias presentes no tráfego de dois *backbones* da internet. Os algoritmos foram capazes de classificar diferentes anomalias como ataques de DoS, varredura de portas e varredura de redes em diferentes classes. Embora os resultados obtidos sejam importantes, os autores não tiveram a intenção de criar um modelo para a classificação automática de anomalias detectadas por um NIDS. Seu principal objetivo foi a validação da factibilidade da classificação das anomalias em diferentes classes. Entretanto, os autores não avaliaram numericamente a precisão de seus resultados e também não consideraram as falsas anomalias (i.e. falsos positivos) que surgem em situações reais.

No trabalho (TELLENBACH et al, 2011) os autores utilizaram o método supervisionado SVM (do inglês SVM: *Support Vector Machines*) para a classificação das anomalias, obtendo uma *Acurácia Global* de classificação de até 85%. A principal contribuição apresentada pelos autores consiste na classificação de anomalias com diferentes intensidades. Uma vez que ataques (e.g. DoS) podem ocorrer em diferentes intensidades, o sistema de classificação deve ser versátil o suficiente para considerar tais variações. No entanto, para a avaliação da classificação os autores assumiram um detector perfeito, o que não condiz com a realidade, já que a detecção de um número elevado de falsos positivos é uma das características dos NIDS baseados em anomalia.

Os autores em (PAREDES-OLIVA et al, 2012) também se utilizaram de métodos supervisionados conhecidos como árvores de decisão para a classificação de anomalias relacionadas a ataques de varreduras (Scans) e negação de serviço (DoS). Neste trabalho destaca-se a alta *Acurácia Global* acima de 98% na classificação dos dados. Neste trabalho os autores estabelecem o conceito de suporte mínimo (*minimum support*) para que um evento, potencialmente uma anomalia, seja eleito para a classificação. A

justificativa dos autores é que um analista não tem condições de analisar todas as anomalias detectadas. No entanto, ao se limitar as anomalias que são processadas pode-se perder eventos importantes em uma rede. Além disso, os autores também não exploram a capacidade da classificação de anomalias para a redução do número de falsos positivos, o que aumentaria a precisão final de detecção.

A hipótese de que a classificação pode ser útil na identificação dos falsos positivos e consequentemente no aumento da precisão na detecção das anomalias foi elencada pelos autores em (FERNANDES; OWEZARSKI, 2009). Neste trabalho realiza-se a classificação das anomalias por meio de assinaturas, que precisam ser geradas para cada anomalia que se deseja classificar. Os autores argumentam que é possível a geração de assinaturas para cobrir a maioria das anomalias verdadeiras, assim, um evento detectado como anômalo que não combinar com nenhuma das assinaturas geradas pode ser considerado um falso positivo. Apesar de ser uma hipótese válida, os autores não a validam e não mostram o quão efetivamente essa classificação pode contribuir para o aumento da precisão de detecção das anomalias.

No trabalho (ZHANG et al, 2015), os autores apresentam um método para a detecção de anomalias de rede com o uso de técnicas de detecção de *outliers.* O modelo desenvolvido se caracterizou pela detecção de um alto número de falsos positivos. Com o objetivo de se reduzir estes falsos alertas, os autores aplicaram o conceito de classificação de anomalias como uma segunda camada de análise, de forma que uma anomalia detectada é considerada um falso positivo sempre que não demonstra um nível mínimo de similaridade com qualquer das classes representantes dos ataques verdadeiros. Nesta classificação, as classes são definidas por subespaços aberrantes (do inglês *Outlying Subspaces*) que são gerados pelo método de busca MOGA (do inglês *Multi-Objective Genetic Algorithm*), que encontra os melhores subespaços inerentes a um conjunto de treinamento. Com isso, os autores obtiveram uma redução de aproximadamente 19% na *TFP*, comprovando a potencialidade da classificação de anomalias também para a redução de falsos positivos.

Assim, em consideração aos trabalhos discutidos, o trabalho ora apresentado discute um modelo de classificação que além de ajudar na rápida identificação das anomalias, também considera a habilidade que a classificação de anomalias possui para a redução de falsos positivos erroneamente detectados pelos NIDSs. Para tal, pretende-se utilizar de métodos de agrupamento, ainda não utilizados para este fim. Esta escolha foi inspirada por trabalhos de classificação de tráfego legítimo, discutidos na seção a seguir.

3.3 Algoritmos de agrupamento na classificação de dados

Como visto na seção anterior, somente o trabalho apresentado em (LAKHINA et al, 2005) se utilizou de métodos de agrupamento para a classificação das anomalias, sendo que a maioria dos trabalhos se fundamentou somente em métodos supervisionados. No entanto, os algoritmos de agrupamento já mostraram bons resultados na área de classificação de tráfego legítimo, uma área bastante correlata à classificação de anomalias cujos resultados também são de grande valia ao tema pesquisado.

Existem muitos trabalhos que tratam da classificação de tráfego legítimo de rede. Estes trabalhos, em sua maioria, são baseados na análise de portas (*port-based*), na análise de *payload* (*payload-based*) ou em algoritmos de aprendizagem de máquina baseados em informações de tráfego TCP/IP.

Os autores Nguyen e Armitage em seu trabalho (NGUYEN; ARMITAGE, 2008) fazem uma extensa revisão sobre o uso de algoritmos de aprendizagem de máquina para classificação de tráfego de rede. Os autores conseguiram mostrar a eficácia destes algoritmos em comparação com os métodos mais tradicionais *(port-based* e *payload-based*). Dentre os algoritmos de aprendizagem de máquina, temos aqueles que se utilizam de métodos supervisionados ou não-supervisionados, a principal diferença entre estes métodos é quanto aos dados de entrada para o treinamento, enquanto que

nos métodos supervisionados os dados precisam ser previamente categorizados, os métodos não-supervisionados não possuem este requisito.

A literatura nos mostra o potencial dos métodos não supervisionados para a classificação de tráfego. O trabalho de Zander et al. (ZANDER et al, 2005) se destaca por conseguir classificar oito diferentes classes de tráfego por meio do algoritmo AutoClass. Em (ERMAN et al, 2006a) os autores fazem uma comparação entre o AutoClass e o método supervisionado Naive Bayes, a conclusão mostrou que o AutoClass superou em cerca de 9 pontos percentuais o algoritmo Naive Bayes na classificação de tráfego de rede em termos de *Acurácia Global*. Em outro trabalho dos mesmos autores (ERMAN et al, 2006b) foi feita a classificação de tráfego de rede com três diferentes algoritmos não supervisionados, foram eles: AutoClass, K-Médias e DBSCAN, e todos apresentaram bom desempenho na classificação dos dados, com destaque ao AutoClass que mostrou a maior *Acurácia Global* dentre os demais algoritmos.

Como pôde ser visto, o algoritmo AutoClass desempenhou um papel bastante importante com relação à classificação de tráfego legítimo de rede, mostrando o grande potencial dos algoritmos de agrupamento. Inspirado por esses resultados, elencou-se o AutoClass juntamente com outras abordagens não supervisionadas como alvos de estudo para a classificação de tráfego não legitimo (i.e. anomalias).

Além do AutoClass, esta pesquisa também devotou atenção ao classificador OPF. Trata-se de um recente algoritmo que emprega tanto o método supervisionado como o não supervisionado. Este tem sido aplicado em diferentes áreas da ciência e se mostrado bastante promissor. Para este trabalho destaca-se as aplicações do OPF no problema binário da detecção de anomalias. No trabalho (PEREIRA et al, 2012), os autores compararam a versão supervisiona do OPF com os algoritmos SVM-RBF, um classificador Bayesiano, e a rede neural SOM para a detecção de anomalias. Nos experimentos, o OPF demonstrou resultados similares aos métodos tradicionais, mas se destacou ao ser a abordagem mais rápida, considerando os tempos de treinamento e classificação. A agilidade dos métodos é um

grande trunfo na detecção de anomalias, pois permite a detecção em tempo real, mesmo em ambientes com maior carga de tráfego.

Os autores em (COSTA et al, 2015) aplicaram a versão não supervisionada do OPF (OPFC) pela primeira vez ao problema de detecção de intrusão. O OPFC foi comparado aos métodos K-médias e SOM. Nos testes o OPFC se mostrou um método eficaz ao demonstrar resultados superiores de classificação com 4 dos 8 conjuntos de dados utilizados para validação.

Devido a esses bons resultados, esta pesquisa também contemplou a análise da versão não-supervisionada do OPF no problema multiclasse da classificação de anomalias. Além do AutoClass e o OPFC, considerou-se também o tradicional K-médias por ser amplamente utilizado em comparações com algoritmos de agrupamento. Além do K-médias, também se elencou o X-médias, uma de suas versões melhoradas que também já mostrou bons resultados na análise de anomalias de rede (AHMED; MAHMOOD, 2014).

Portanto, este trabalho também contribui com uma análise comparativa de diferentes algoritmos de agrupamento, ainda não utilizados para a classificação de anomalias. Esta análise é inédita até onde se sabe e tem como objetivo elencar métodos eficazes para o auxílio de outros pesquisadores em suas pesquisas neste tema.

3.4 Considerações finais

Neste capítulo foram apresentados alguns trabalhos relacionados à classificação de anomalias e ao uso de métodos de agrupamento para a classificação em diferentes domínios. Os conceitos aprendidos e os problemas em aberto apresentados nestes artigos foram essenciais para a delimitação do escopo desta pesquisa. O desenvolvimento do trabalho proposto é apresentado em detalhes a partir do próximo capítulo.

CAPÍTULO 4 - Metodologia

4.1 Considerações iniciais

Este capítulo descreve a metodologia para o desenvolvimento da pesquisa apresentada nesta dissertação. Pretende-se enriquecer o estado da arte na área de classificação de anomalias com uma investigação de algoritmos de agrupamento que sejam eficazes para este fim. Ainda, espera-se explorar a capacidade que a classificação de anomalias possui para aumentar a *Precisão* de detecção das anomalias. Este capítulo é dividido em três seções, incluindo esta. Na Seção 4.2 são explicados os objetivos pretendidos e os resultados esperados. Na Seção 4.3 é mostrada a arquitetura e funcionamento do modelo proposto, pelo qual se espera alcançar os objetivos descritos.

4.2 Objetivos

Como já discutido no capítulo anterior, os NIDS baseados em anomalia receberam grande atenção nos últimos anos. No entanto, grande parte das pesquisas sobre detecção de anomalias focam em desafios como a definição de fronteiras para a delimitação do comportamento normal, o tratamento de ruídos que podem levar a dados normais serem considerados anômalos, a disponibilidade de conjuntos de dados categorizados para o treinamento das

ferramentas, dentre outros (CHANDOLA; BANERJEE; KUMAR, 2009). Neste contexto, poucos são os trabalhos que além da diferenciação entre comportamento anômalo e normal, focam na distinção entre os diferentes tipos de anomalia (PAREDES-OLIVA et al, 2012). A não distinção e identificação das anomalias é bastante prejudicial, pois exige que este processo seja feito manualmente pelos analistas, o que pode ser bastante dispendioso.

Neste sentido, justifica-se o empenho em pesquisas que buscam, por meio de classificação, a rápida identificação das anomalias detectadas. Por este motivo este trabalho propõe um modelo para automaticamente classificar as anomalias detectadas por um NIDS baseado em anomalia. Pretende-se contribuir com o estado da arte deste tema com a investigação de algoritmos de aprendizagem de máquina não supervisionados, também conhecidos como métodos de agrupamento, ainda não utilizados para este fim. Com isso, espera-se validar os algoritmos que mostrarem o melhor desempenho para a classificação proposta. Esta comparação entre diferentes modelos é útil, pois orienta a escolha de um algoritmo eficiente para compor o modelo proposto e também pode ajudar futuros pesquisadores neste tema.

Além da identificação das anomalias detectadas, este trabalho também pretende explorar a capacidade de redução de falsos positivos que a classificação de anomalias pode apresentar. Um modelo de classificação, quando bem treinado, pode classificar o alerta de uma anomalia entre uma das classes válidas de ataque ou pode também classificá-lo entre classes representativas de tráfego legítimo. Dessa forma, um comportamento legítimo, que tenha sido erroneamente detectado como uma anomalia, pode ser identificado no momento da classificação quando este for classificado entre os dados legítimos. Dessa forma, cria-se uma segunda camada de decisão que pode ajudar na filtragem de falsos alertas.

Portanto, o objetivo final é a criação de um modelo de classificação que auxilie na identificação das anomalias e também dos falsos positivos erroneamente classificados. Com isso em mãos, os analistas de segurança possuem mais recursos para suas análises, uma vez que a classificação pode permitir o uso de ferramentas avançadas de correlação (MIRSHAHJAFARI;

GHAVAMNIA, 2014) e também ajudar na tomada de contramedidas em tempo hábil (PAREDES-OLIVA et al, 2012).

4.3 Arquitetura e funcionamento

Para se atingir os objetivos descritos, foi arquitetado um modelo para a classificação automática das anomalias detectadas por um NIDS baseado em anomalia. Este modelo, ao qual deu-se o nome de *Anomalies Identifier* (AnoID), conta com os componentes observados na Figura 4-1. Como pode ser visto, o AnoID é composto por dois componentes principais, a saber: *Anomalies Detector* (AD) e *Anomalies Classifier* (AC). O AD consiste no NIDS baseado em anomalia responsável pela detecção das anomalias e emissão dos alertas, os quais deverão ser classificados pelo AC. O AC possui três subcomponentes: *Features Extractor* (FE), *Clustering Algorithm* (CA) e o *Class Reader* (CR). O CA é o principal componente deste modelo, pois compreende o algoritmo de agrupamento utilizado para a classificação. A fim de se elencar o melhor algoritmo para a classificação proposta, esta pesquisa avaliou os seguintes algoritmos: AutoClass, OPFC, K-médias e X-médias. Todos esses implementam métodos de agrupamento, assim como inicialmente proposto.

Como visto na Seção 2.5, os métodos de agrupamento realizam a descoberta automática dos grupos (ou classes) naturais de um conjunto de dados de treinamento. Tais grupos representam, neste projeto, as possíveis classes de ataque ou tráfego legítimo em que as anomalias podem ser classificadas. Para isso, é necessária a realização de um mapeamento prévio para se atribuir significado a cada um dos grupos encontrados, um processo típico dos métodos de agrupamento (NGUYEN; ARMITAGE, 2008).

Neste trabalho, o mapeamento dos grupos é realizado de acordo o evento majoritariamente classificado em cada um deles na etapa de treinamento. Quanto maior a fração de instâncias de um único evento classificado em um certo grupo, maior a sua pureza. Esta pureza é medida formalmente pela HI, explicada na Seção 2.4. Quanto maior o valor de HI

obtido por um grupo, maior a sua capacidade de classificar o evento majoritariamente agrupado a ele e, portanto, tal grupo recebe o rótulo do evento em questão. Os grupos encontrados na etapa de treinamento, após mapeados para as classes de interesse, são utilizados pela etapa de classificação, sendo que as anomalias receberão o rótulo do grupo a que forem classificadas.

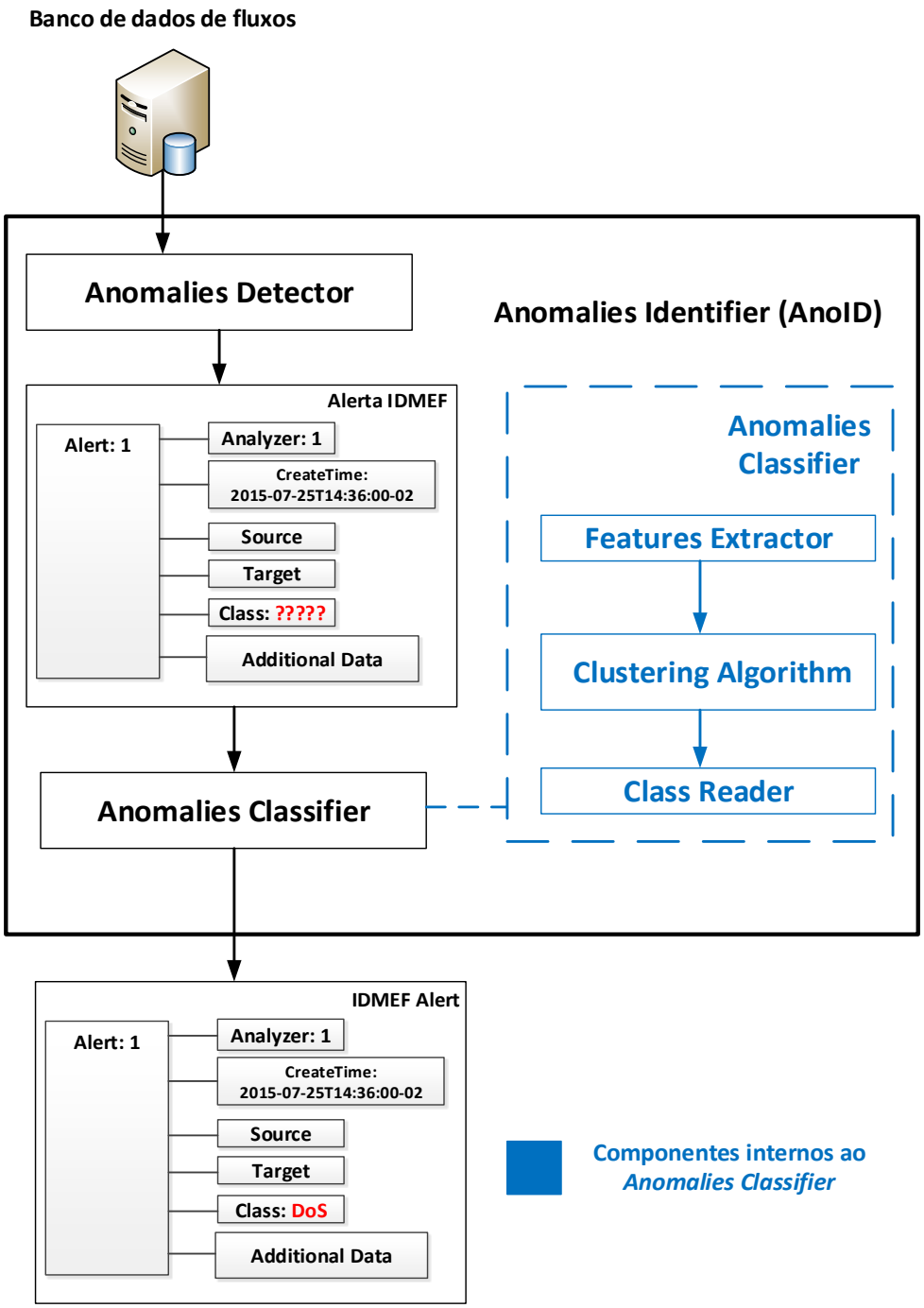


Figura 4-1. Arquitetura do modelo proposto.

Com a etapa de treinamento cria-se então um modelo de classificação. Neste modelo, uma classificação é correta quando um alerta de uma anomalia, a princípio desconhecida, é corretamente classificado no grupo mapeado para o ataque que o gerou. Além das verdadeiras anomalias, busca-se também a classificação das falsas anomalias detectadas pelo AD. As falsas anomalias representam a má classificação de tráfego legítimo como anomalia (i.e. falso positivo). Uma vez que se consiga classificar esses casos em meio aos grupos mapeados para o tráfego legítimo, é possível poupar o tempo que os analistas dispenderiam na análise destes.

Num problema de classificação os dados devem ser modelados na forma de instâncias, que são definidas por um conjunto de características descritoras. Para este trabalho, definiu-se características que resumem o tráfego gerado por um determinado *host* no ambiente monitorado em um dado momento. Para a fase de treinamento, as instâncias são representadas por conjuntos de fluxos que são gerados pelo agrupamento do tráfego por cada endereço de origem presente em períodos de 5 minutos. Dessa forma, cada instância é representada por um certo conjunto de fluxos associado ao um certo endereço de origem encontrado. Do conjunto de fluxos que define uma instância, são extraídas algumas características que descrevem o comportamento apresentado pelo endereço de origem em questão. Estas características são apresentadas na

Tabela 4-1.

Tabela 4-1. Características escolhidas para a classificação no AC.

Nome	Descrição
total_src_pkts	Número de pacotes enviados.
total_src_bytes	Número de *bytes* enviados.
timerate	Taxa de conexões que não são requisições HTTP ou HTTPS que tiveram menos de 15 segundos de duração.
nullrate	Taxa de fluxos sem respostas do destino.
rstrate	Taxa de fluxos com a *flag* RST ativada.
distinct_dstports	Taxa de portas distintas de destino acessadas.
distinct_dstwkps	Taxa de portas distintas (*well known port*) de destino acessadas.
distinct_dstaddrs	Taxa de IPs destino distintos acessados.

Abaixo é apresentada uma breve justificativa para as características escolhidas:

- *total_src_pkts*: contribui, principalmente, para a separação do tráfego legítimo do anômalo, já que a grande maioria dos ataques detectáveis por fluxos apresentam altas taxas de pacotes enviados, especialmente os ataques de varredura de vulnerabilidades web, dicionário e negação de serviço.
- *total_src_bytes*: tem como papel, além de separar o tráfego anômalo do legítimo, representar diferenças entre os ataques, visto que o ataque de varredura de vulnerabilidades web, além de uma grande quantidade pacotes, também envia uma grande quantidade de bytes devido a carga útil dos pacotes enviados, o que não acontece de forma tão intensa com o ataque de DoS, por exemplo.
- *distinct_dstaddrs*: contribui fortemente para a identificação de eventos que impliquem várias conexões a variados endereços de destino de uma rede, como exemplo destaca-se o ataque de varredura de redes.
- *timerate*: especialmente elencada para a identificação dos ataques de dicionário e varredura de portas que realizam uma grande quantidade conexões de curta duração em um pequeno espaço de tempo.
- *rstrate:* característica especialmente útil na identificação dos ataques de varredura de portas, visto que sempre que uma porta fechada é varrida, retorna-se um pacote com a *flag* RST ativada ao endereço que originou a conexão. Este comportamento eleva o valor desta taxa em comparação com os demais eventos.
- nullrate: esta característica contribui para a classificação de eventos como a varredura de redes, que apresenta várias requisições a *hosts* inexistentes, e a negação de serviço, pois, uma vez que o *host* alvo se encontra indisponível é incapaz de responder às novas requisições recebidas.

- *distinct_dstports*: Esta característica é útil para a identificação de eventos que realizam conexões a uma grande quantidade de portas distintas, como no caso do ataque de varredura de portas.
- *distinct_dstwkps:* característica importante para promover diferenciação entre os ataques de varredura de portas, que não realizam acessos a somente portas privilegiadas, e outros ataques que consistem na exploração de serviços que executam sobre essas portas, como por exemplo o ataque de negação de serviço contra um servidor Web ou um ataque de dicionário contra um servidor FTP.

Como visto, estas características foram selecionadas por sua capacidade de conjuntamente representar de forma distinta o comportamento de diferentes tipos de tráfego. Estas podem ser facilmente extraídas de protocolos de sumarização como o *Netflow* (CLAISE, 2004), um protocolo amplamente utilizado em ambientes operacionais (PAREDES-OLIVA et al, 2012).

Na etapa de classificação, as instâncias são os alertas, representando as anomalias que se queira classificar. Para tal, cada alerta deve também conter as informações apresentadas na

Tabela 4-1, referentes ao comportamento desempenhado pelo endereço de origem indicado no alerta. Essas características são extraídas de cada alerta pelo FE, que então gera os arquivos de entrada necessários para a classificação que é realizada no CA. A fim de se prover uma interoperabilidade maior, o AnoID foi arquitetado para processar alertas recebidos no protocolo IDMEF (do inglês IDMEF: *Intrusion Detection Message Exchange Format*), descrito pelo RFC 4765. O IDMEF é um padrão que define a forma como os IDSs em geral podem reportar seus alertas. Este é um padrão bastante dinâmico que pode permitir desde a padronização de um simples alerta, com poucas informações, até um alerta mais robusto que contemple informações extras. Dessa forma, qualquer NIDS que exporte seus alertas no padrão IDMEF e relate as informações da

Tabela 4-1 para os eventos detectados pode agir como o AD do modelo proposto.

Ao término da classificação é ativado o CR, o qual tem como missão a leitura das informações geradas pelo CA e a determinação da classe do alerta de acordo com o mapeamento previamente realizado. Esta classe pode ser uma taxonomia de ataque (e.g. DoS, varredura de portas ou ataque de dicionário) ou uma indicação de tráfego legítimo. Neste trabalho, este último caso indica que o alerta recebido é na verdade uma falsa anomalia erroneamente detectada. No entanto, em situações específicas, este caso também pode representar uma anomalia verdadeira não conhecida pelo modelo de classificação, sendo esta uma hipótese não contemplada por esta pesquisa.

Portanto, como entendido na Figura 4-1 e na descrição acima, vê-se que o produto gerado pelo AnoID é um alerta no formato IDMEF com sua classe definida. A classe da anomalia detectada é de grande importância, uma vez que ferramentas avançadas de correlação de alertas e construção de cenários necessitam dessa informação para operar. Ademais, a informação clara sobre a causa de uma anomalia permite a tomada de contramedidas de forma mais rápida e eficiente. Somando-se isto à diminuição da quantidade de falsos positivos, pode-se diminuir as chances de se aplicar uma contramedida contra um usuário legítimo, o que poderia ocasionar uma violação do Acordo de Nível de Serviço (do inglês SLA: *Service Level Agreement*) (CHUNG et al, 2013) entre provedores e clientes. Assim, o AnoID seria também de grande valia em ambientes que respondem de forma automática aos eventos detectados.

4.4 Considerações finais

Neste capítulo foi descrito o AnoID, o *framework* proposto para a identificação de anomalias baseado na integração de um NIDS com um classificador baseado em métodos de agrupamento. No próximo capítulo é discorrido sobre os testes e resultados obtidos com o AnoID, utilizando-se de diferentes métodos de agrupamento e também abordagens de treinamento.

CAPÍTULO 5 - Testes e Resultados

5.1 Considerações iniciais

Este capítulo é reservado para a apresentação dos testes realizados e os resultados obtidos com a abordagem proposta nesta pesquisa. Na Seção 5.2 se discorre sobre a geração do conjunto de anomalias a ser utilizado para os testes de validação, além dos dados utilizados para o treinamento dos algoritmos. Na Seção 5.3 são apresentados e discutidos os resultados obtidos com o algoritmo AutoClass, na Seção 5.4 com o algoritmo OPFC, na Seção 5.5 com o algoritmo K-Médias e na Seção 5.6 com o algoritmo X-Médias. Na Seção 5.7 é feita uma análise comparativa entre o desempenho dos diferentes algoritmos. Na Seção 5.8 são apresentados os resultados relacionados ao impacto da redução de falsos positivos por meio da classificação de anomalias. Por fim, na Seção 5.9 são feitas as considerações finais acerca dos resultados obtidos.

5.2 Geração dos dados para validação

Esta seção trata sobre a geração dos dados utilizados na validação do modelo de classificação proposto. Esses dados incluem o tráfego de rede utilizado para o treinamento dos algoritmos de agrupamento, além das anomalias para a validação da classificação de cada algoritmo.

Esta pesquisa compartilha das preocupações demonstradas em (RINGBERG; ROUGHAN; REXFORD, 2008) com relação aos dados para a avaliação dos sistemas de detecção baseados em anomalia. Dessa forma, evitou-se o uso de tráfego real, manualmente categorizado, para a representação das anomalias e se optou pela simulação, assim como sugerido pelos autores. O uso de dados manualmente classificados para a validação de sistemas apresenta uma série de dificuldades, como a sujeição a falhas na classificação manual, a dificuldade em se obter dados em uma magnitude necessária para a validação dos sistemas, o compartilhamento desses dados para permitir a reprodutibilidade da pesquisa (e.g., preocupações com dados proprietários e privacidade dos usuários), entre outros. No entanto, sabe-se que em ambientes reais os distúrbios são maiores e, com isso, a taxa de falsos positivos tende a ser maior, sendo este o principal preço ao se utilizar de dados sintéticos.

Para a geração de todos os dados utilizados nos testes, utilizou-se o ambiente de coleta de dados descrito na seção a seguir.

5.2.1 Ambiente de coleta de dados

Para a geração dos dados, criou-se um ambiente de coleta composto por: 1) um ambiente virtual construído por meio do software KVM (do inglês KVM: *Kernel-based Virtual Machine*) (KVM, 2016) para a simulação dos ataques e 2) a rede interna do Laboratório ACME! de pesquisa em Cibersegurança, onde esta pesquisa foi realizada, para a coleta de dados legítimos gerados por usuários reais. Uma vez que se pretende avaliar a classificação das falsas anomalias, é necessário também que haja dados de cunho não anômalo para que se possa avaliar as detecções errôneas do NIDS escolhido para compor o AD. Um esquema deste ambiente pode ser visto na Figura 5-1.

Como visto na Figura 5-1, o ambiente de coleta de dados é composto por três principais redes: a rede atacante, a rede alvo e a rede interna do laboratório ACME!. As setas contínuas indicam conexões de rede, enquanto

que as trastejadas indicam a transição lógica de informações que trafegam pela rede, mas tem como destino *hosts* específicos.

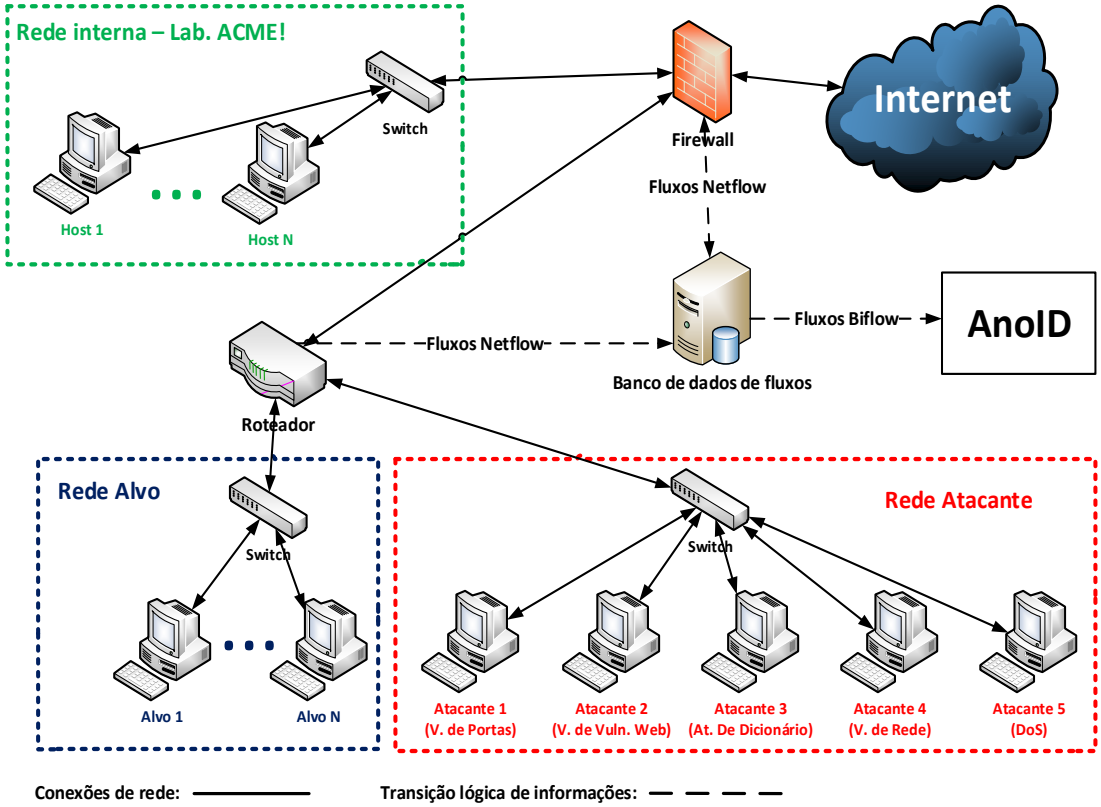


Figura 5-1. Ambiente de coleta de dados.

Rede Atacante

A rede atacante foi criada em ambiente virtual e é composta por cinco máquinas, de forma que cada máquina foi configurada para a execução de um ataque em específico. Esta configuração foi essencial para a criação do conjunto de dados, uma vez que qualquer base para avaliação de modelos de classificação precisa ser bem categorizada. Com a execução de cada ataque com um endereço de origem distinto, foi possível a categorização de cada fluxo anômalo no ambiente de acordo com a categoria de cada ataque. Este processo foi essencial para o mapeamento descrito na Seção 4.3.

Os ataques considerados para a avaliação do modelo proposto são explicados na Seção 2.2, já nesta seção é discorrido apenas sobre as características de execução de cada um deles, isto é, ferramentas e técnicas.

- Varredura de portas (V. de portas): para a execução deste ataque se utilizou da ferramenta Nmap (do inglês “*Network Mapper*”) (Nmap, 2016), uma das mais populares na categoria de varreduras de segurança. Para a varredura de portas se escolheu a modalidade TCP SYN, que consiste no envio de pacotes com a *flag* SYN ativada à várias portas de um determinado *host*. No caso de portas que estejam abertas, será recebida uma resposta SYN/ACK, já o recebimento de uma resposta RST é um indicativo que a porta está fechada.
- Varredura de vulnerabilidades *Web* (V. de Vuln. Web): para esta categoria de ataques se utilizou do *Arachni*, um framework *Open Source* de testes de segurança em aplicações Web. Esta ferramenta foi escolhida devido a sua grande versatilidade no que diz respeito aos testes que desempenha. O *Arachni* realiza checagens ativas que envolvem o envio de dados aos campos de entrada da aplicação web para se encontrar falhas de Injeção de SQL, CSRF (do inglês CSRF: *Cross-site request forgery*) e injeções de código no geral. Além disso, também podem ser desempenhadas algumas checagens passivas, como a procura pela existência de arquivos, pastas e assinaturas. Para os testes se utilizou o *Arachni* em seu modo padrão, que realiza todas as checagens disponíveis, maiores informações podem ser consultadas em (Arachni, 2016).
- Ataque de dicionário (At. de Dicionário): a realização do ataque de dicionário se deu com a ferramenta Hydra (Hydra, 2016), também bastante popular para este fim. Esta ferramenta se destaca por sua capacidade multitarefa em realizar as tentativas, sendo compatível com mais de 50 protocolos, incluindo *telnet, ftp, http, ssh* etc. Nos testes, realizou-se o ataque contra um servidor ftp da Rede Alvo, para isso se utilizou um dicionário com 233.722 senhas com 9 tarefas em concorrência.
- Varredura de redes (V. de Redes): este ataque também foi realizado com a ferramenta Nmap na modalidade *Ping Scan*,

onde o Nmap envia requisições ICMP, TCP SYN para a porta 443, TCP ACK para a porta 80 e mensagens ICMP *timestamp* para todos os *hosts* em um dado intervalo.

- Negação de serviço (DoS): este ataque foi realizado por uma aplicação em C que estabelece diversas conexões TCP na porta 80 de um servidor HTTP, realizando a negação de serviço pelo esgotamento dos recursos do servidor. Este ataque foi configurado para ser executado durante 30 minutos por série.

Rede Alvo

Todos os ataques originados na rede atacante são realizados contra as máquinas da rede alvo, a qual é composta por máquinas com diversos serviços em execução, constituindo os alvos dos ataques já explicados. A rede alvo também foi configurada virtualmente, no mesmo ambiente utilizado para a virtualização da rede atacante.

Rede do Laboratório ACME!

Como já mencionado brevemente, a rede interna do Laboratório ACME! foi utilizada para a coleta de dados legítimos (i.e. de cunho não anômalo). Esses dados são de grande valia para esta pesquisa, posto que a redução de falsos positivos é também frente deste trabalho.

Esta rede possui características especiais que permitiram uma coleta bastante fiel dos dados. Trata-se de um ambiente bastante controlado, protegido por firewall e utilizado por especialistas, de forma que se garante um alto grau de confiança no cunho não anômalo das informações coletadas. Em contrapartida esta é uma rede com um tráfego bastante diverso, envolvendo tráfego http, smtp, bate-papo, acesso remoto, além de outras operações comuns realizadas por usuários legítimos.

Coleta dos dados

Como também pode ser visto na Figura 5-1, todos os dados foram coletados na forma de fluxos de dados no formato *Netflow* por meio de exportadores posicionados em locais estratégicos da rede. Como

exportadores utilizou-se o *fprobe* (Fprobe, 2016), no roteador entre as redes alvo e atacante, e o exportador nativo do Ubiquiti EdgeRouter™ PRO, utilizado como roteador e Firewall para as redes do laboratório ACME!. Dessa forma, os fluxos coletados sumarizam todos os dados trocados entre as redes alvo e atacante e também entre as redes do laboratório ACME!. Os fluxos foram coletados pelo *Biflow Collector* (PROTO, 2011), este é um coletor que recebe os fluxos no formato *Netflow* e os converte para o formato *Biflow*, armazenando-os no banco de dados de fluxos, ilustrado na Figura 5-1.

5.2.2 Conjunto de dados para treinamento dos algoritmos de agrupamento

O primeiro passo ao se trabalhar com algoritmos de aprendizagem de máquina é a definição de uma boa base de dados para o treinamento dos métodos. Para esta pesquisa, elencou-se duas diferentes hipóteses para isso. O objetivo foi a avaliação da melhor abordagem para se realizar a classificação de anomalias com ênfase na redução de falsos positivos, são elas:

1. **Abordagem Centrada em Ataques (CA):** Nesta abordagem o treinamento dos algoritmos é realizado somente com os dados obtidos pelo tráfego entre as redes alvo e atacante. Isso significa que a grande maioria dos dados são anômalos, provenientes da simulação dos ataques. Entretanto, como em toda rede, este tráfego também compreende uma pequena porcentagem de dados de controle como mensagens *broadcast*, consultas DNS, consultas NTP, entre outros.

   Neste caso espera-se que os algoritmos criem grupos muito bem definidos para os ataques simulados, com os quais espera-se uma melhor classificação das anomalias verdadeiras. Além dos grupos representativos dos ataques, também se espera que sejam gerados grupos para agruparem os dados de controle trocados em meio aos ataques. Assim, espera-se que as falsas anomalias detectadas sejam classificadas em meio aos grupos de controle, uma vez que os grupos de ataques serão

muito bem definidos e, portanto, com pouca tendência em classificar dados que sejam de aplicações legítimas.

Deste modo espera-se um alto desempenho na classificação dos ataques, mas um desempenho reduzido na classificação das falsas anomalias. Ou seja, é esperado que as classes de controle não apresentem uma representatividade suficiente para a classificação das falsas anomalias, isso devido à pouca quantidade de informação sobre aquilo que se considera legítimo.

2. **Abordagem Centrada em Falsas Anomalias (CFA):** Nesta abordagem, diferentemente da abordagem CA, há de se realizar o treinamento dos algoritmos tanto com os dados anômalos provenientes da simulação dos ataques, como também com dados provenientes do tráfego legítimo gerado pelos usuários do Laboratório ACME!. Dessa forma, almeja-se a geração de grupos com uma boa representatividade para o tráfego não anômalo e, com isso, a obtenção de uma performance maior que aquela obtida na abordagem CA na classificação das falsas anomalias.

   No entanto, com o aumento considerável na quantidade e na heterogeneidade do conjunto de dados de treinamento, proporcionado pela adição dos dados relativos ao tráfego da rede do laboratório ACME!,, é esperado um aumento na complexidade da classificação, ocasionando uma redução no desempenho da classificação das anomalias verdadeiras.

Com essas duas abordagens de treinamento, a ideia é se obter dois modelos de classificação distintos. O modelo baseado na abordagem CA deverá ser bastante preciso com relação à classificação dos ataques em suas classes corretas ao custo de uma eficácia reduzida na identificação das falsas anomalias. Em contrapartida, almeja-se que o modelo baseado na abordagem CFA se mostre altamente eficaz na identificação das falsas anomalias ao custo de uma leve redução em sua capacidade de classificar as anomalias verdadeiras. O objetivo disto é analisar as vantagens e desvantagens de

diferentes abordagens, buscando-se modelos que possam se adequar a diferentes realidades.

Com relação aos dados anômalos, utilizados por ambas as abordagens, cada máquina da rede atacante foi configurada para executar seus ataques durante um período de 24 horas, realizando-se uma pausa de 5 minutos entre o término de uma série e o início da próxima. Esta pausa foi inserida para se evitar que os ruídos gerados por uma série de ataques influenciem os ruídos da próxima série, o que poderia gerar um treinamento viciado, impedindo a classificação de ataques executados isoladamente e não em séries como ocorreu no treinamento. No tocante aos dados legítimos utilizados pela abordagem CFA, foram coletados todos os dados produzidos pelos usuários do laboratório ACME! no mesmo período de 24 horas de realização dos ataques.

No que tange ao período de 24 horas para a coleta dos dados, escolheu-se o intervalo compreendido entre os dias 09 e 10 de novembro de 2015, iniciando-se às 11 da manhã do dia 09; ambos dias úteis do mês de novembro, de modo que o tráfego na rede interna do laboratório ACME! foi pleno. Todos os fluxos gerados neste período foram armazenados no banco de fluxos e processados para a geração das instâncias.

O conjunto de dados a ser utilizado na abordagem CA foi gerado a partir do processamento dos fluxos referentes ao tráfego entre as redes alvo e atacante. O resultado desse processamento foi um conjunto com 2.907 instâncias de treinamento.

O conjunto a ser utilizado na abordagem CFA foi gerado a partir da adição das instâncias geradas pelos fluxos da rede do laboratório ACME! ao conjunto utilizado na abordagem CA. Como resultado, se obteve um conjunto com 19.739 instâncias de treinamento, compreendendo tanto amostras dos ataques gerados quanto dos dados legítimos trocados pela rede do laboratório ACME!.

5.2.3 Conjunto de anomalias para validação dos algoritmos de agrupamento

A fim de compor o AD do AnoID, foi escolhido o NIDS descrito em [(BATISTA; CANSIAN, 2011) e (BATISTA, 2012)]. Este é um NIDS baseado na rede neural SOM (*Self-Organizing Maps*) que realiza suas detecções através da análise de fluxos bidirecionais. Para tal, a rede precisa ser treinada de modo a representar o comportamento legítimo do ambiente a ser analisado para então realizar uma detecção baseada em anomalia. Seguindo as características deste tipo de sistema, este NIDS somente identifica se determinado comportamento é anômalo ou não e não fornece maiores detalhes sobre a anomalia detectada, o que o torna um candidato ideal ao modelo proposto.

Para uma validação efetiva dos algoritmos de agrupamento, almejou-se a geração de uma quantidade considerável de alertas, representando tanto anomalias verdadeiras como também falsos positivos. Para tal, o AD foi configurado para monitorar o ambiente de coleta de dados durante 96 horas, compreendendo o período de dias úteis de 14 de dezembro de 2015 a 18 de dezembro de 2015, com início às 10h45 do dia 14. Neste período os ataques foram configurados para serem executados com períodos de intervalo aleatórios entre 5 a 15 minutos entre o final e início da próxima série. Optou-se por esse intervalo aleatório para garantir que o modelo de classificação fosse capaz de lidar com ataques sem uma frequência fixa, como aconteceu na execução dos ataques utilizados na etapa de treinamento. Além dos ataques, o AD também monitorou todo o tráfego trocado na rede interna do laboratório ACME!.

Durante o período de monitoramento o AD analisou 344.091 amostras de fluxo, sendo 20.950 amostras referentes aos ataques simulados e 323.141 amostras referentes ao tráfego legítimo produzido pelos pesquisadores do laboratório ACME!. Uma análise dos resultados obtidos permitiu a construção da matriz de confusão exibida na Tabela 5-1. Observando-se os dados desta matriz nota-se o valor de 316.811 verdadeiros negativos, 2.286 falsos negativos, 6.330 falsos positivos e 18.664 verdadeiros positivos.

Tabela 5-1. Matriz de confusão com os resultados da detecção pelo AD.

	Casos Preditos		Totais
Casos Reais	Tráfego Legítimo	Ataques	
Tráfego Legítimo	**316.811**	**6.330**	**323.141**
Ataques	**2.286**	**18.664**	**20.950**

Com esses dados é possível o cálculo das métricas de desempenho descritas na Seção 2.4. Os valores são apresentados na Tabela 5-2.

Tabela 5-2. Métricas para os resultados de detecção do AD.

Acurácia	***Precisão***	***TVN***	***TVP***
97,50%	74,67%	98,04%	89,09%

Em termos gerais o AD obteve bons resultados, com uma *Acurácia* de 97,50%, valor influenciado pelo alto valor de *TVN* também obtido. Com relação à *TVP* o AD obteve um valor de 89,09%, o que indica uma alta porcentagem de ataques detectados.

No entanto, ao se observar a taxa de *Precisão* obtida, nota-se um valor moderado de 74,67%, inferior aos altos valores obtidos na *Acurácia* e *TVN*, por exemplo. A *Precisão*, em especial, é uma métrica bastante importante para esta pesquisa. Como indicado pela Equação 2.3, esta métrica relaciona os valores de VPs sobre os valores de FPs. Uma vez que VPs e FPs representam os casos positivos detectados pelo AD, o número $VP + FP$ indica a quantidade de alertas emitidos pelo NIDS, que nesse experimento foi de 24.994. Assim, o cálculo: $1 - Precisão$ indica a quantidade de falsos alertas em meio ao conjunto total de alertas. Dessa forma, dos 24.994 alertas detectados 25,33% são falsos. Isso corresponde ao valor de 6.330 alertas que seriam desnecessariamente analisados por um analista ou processados por algum método de correlação.

Além da classificação das anomalias verdadeiras, este projeto também se propõe a identificar os alertas erroneamente detectados, que ocorrem em quantidades elevadas nos métodos baseados em anomalia. Na discussão dos resultados, este conjunto de falsos alertas será denominado falsas anomalias.

O objetivo é classificá-las junto a classes representativas de tráfego legítimo, de modo que não sejam consideradas por analistas e métodos automáticos de análise.

Ao se analisar o conjunto de alertas emitidos, com o conhecimento das características do ambiente de coleta, foi possível construir o gráfico ilustrado na Figura 5-2 com a proporção dos alertas referentes a cada ataque, além das falsas anomalias. Na Figura 5-2 observa-se que os ataques mais duradouros foram aqueles que mais geraram alertas, como por exemplo o ataque de dicionário e o DoS. Além disso, nota-se que o conjunto de falsas anomalias representa cerca de um quarto de todo o conjunto.

Este conjunto com 24.994 alertas constituem a base de anomalias para os testes de validação de todos os algoritmos elencados por esta pesquisa. Os resultados obtidos por cada um deles são apresentados nas seções a seguir.

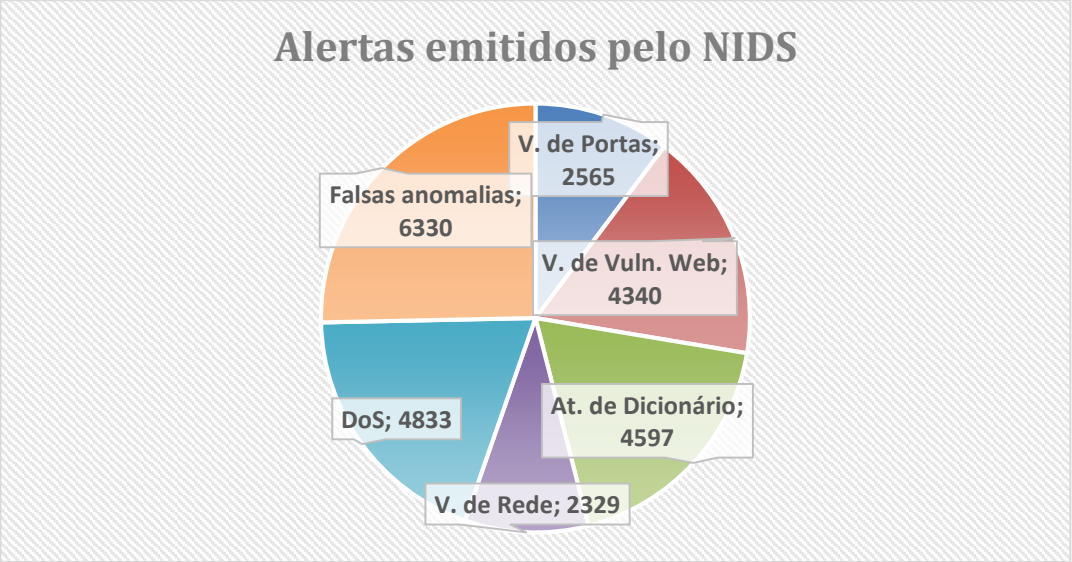


Figura 5-2. Proporção de cada tipo de alerta no conjunto obtido.

5.3 Resultados com o algoritmo AutoClass

Esta seção apresenta os resultados obtidos com o algoritmo AutoClass, descrito na Seção 2.5.2. Sabe-se que a normalização dos dados é uma boa

prática na utilização de algoritmos de aprendizagem de máquina. Entretanto, esta prática artificializa a escala entre as características. Os autores, em (CHEESEMAN; STUTZ, 1996), constataram que essa artificialização tende a prejudicar os modelos de probabilidade do AutoClass, reduzindo seu desempenho geral de classificação. Essa constatação foi de fato verificada em testes preliminares desta pesquisa. Dessa forma, exclusivamente para o algoritmo AutoClass não se utilizou de nenhum tipo de normalização para os dados.

O AutoClass não necessita de parâmetros específicos para executar o treinamento, como por exemplo o K do algoritmo K-médias. Os parâmetros necessários consistem basicamente na configuração de uma condição de parada e os modelos probabilísticos para cada atributo, os quais foram oportunamente discutidos na Seção 2.5.2.

Os experimentos para todos os algoritmos foram feitos considerando-se as abordagens CA e CFA. Na seção a seguir são mostrados os resultados com abordagem CA.

5.3.1 Resultados com abordagem CA

O treinamento do AutoClass com os dados da abordagem CA levou 52 segundos e foram geradas 17 diferentes classes. A disposição dos dados de treinamento em cada uma das classes pode ser vista na Figura 5-3. Nesta imagem, cada barra vertical representa uma classe. As cores em cada classe representam a porcentagem de instâncias de cada evento (i.e., ataques ou falsas anomalias) que foram agrupadas a ela. Quanto mais homogêneo for o conjunto de cores em cada classe, mais instâncias de um único evento foram agrupadas. Por exemplo, classes com uma única cor agruparam instâncias de somente um tipo de evento.

Como explicado na Seção 4.3, cada classe encontrada no treinamento deve ser mapeada para algum evento de acordo com a sua pureza na classificação. No mapeamento para este experimento, como pode ser verificado na Figura 5-3, duas classes receberam o rótulo do ataque de varredura de portas, uma vez que agruparam majoritariamente as instâncias

deste evento. Duas outras classes foram mapeadas para o ataque de varredura de vulnerabilidades web, uma classe para o ataque de dicionário, uma classe para o ataque de varredura de rede, duas classes para o ataque de DoS e nove classes foram mapeadas para falsas anomalias, posto que majoritariamente agruparam as instâncias de tráfego legítimo.

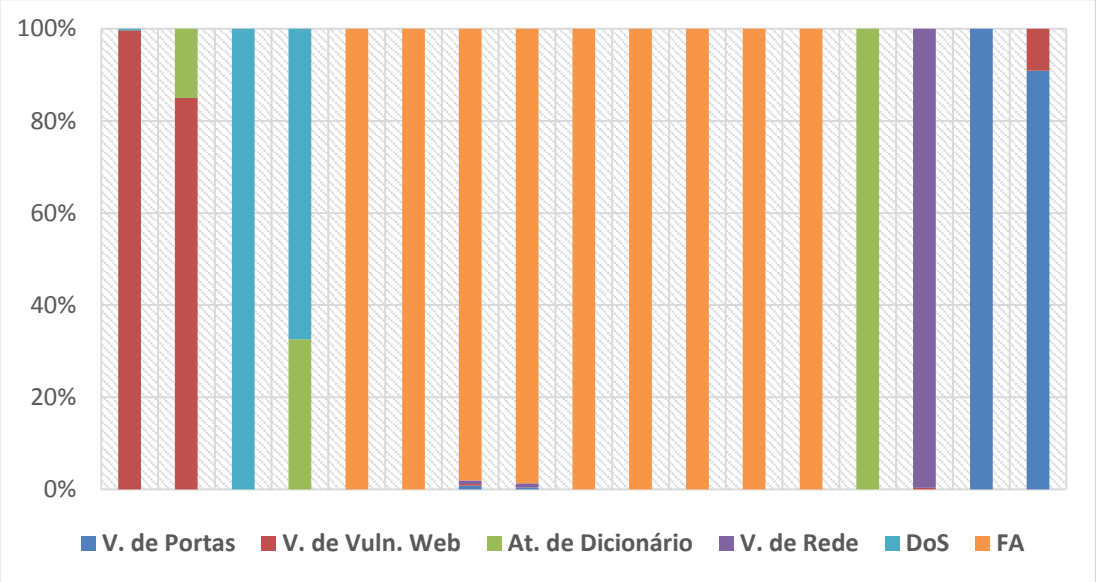


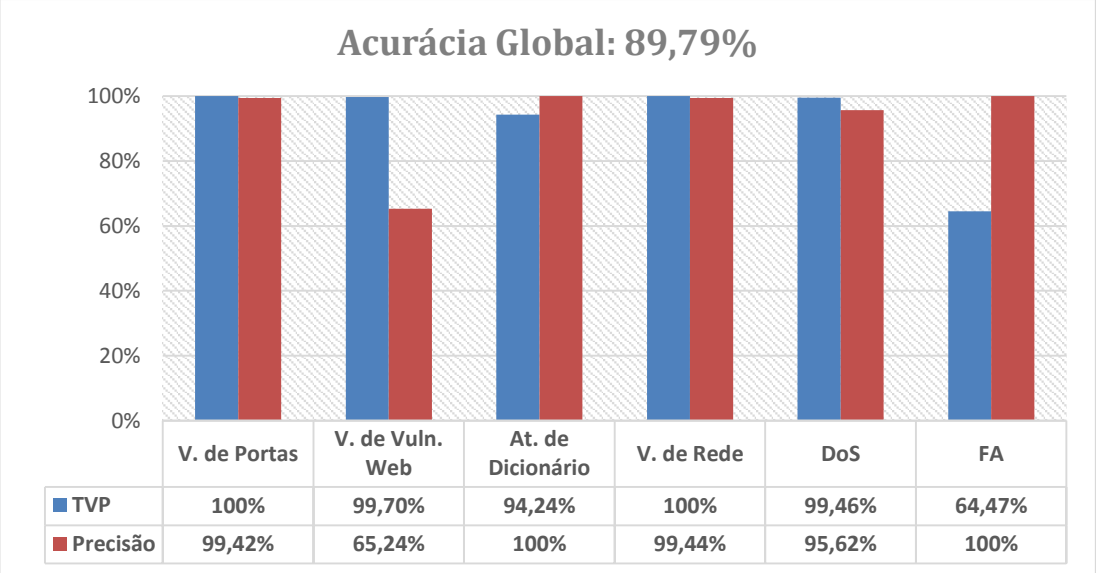
Figura 5-3. Disposição das instâncias nas classes de treinamento.

O valor da HGP, definida na Equação 2.8, calculada a partir da homogeneidade intraclasse de cada classe encontrada no treinamento foi de 98,93% para este experimento. Este índice sugere a capacidade do AutoClass em classificar os eventos de diferentes tipos em diferentes classes, uma característica essencial para a classificação proposta. Este alto valor de HGP pressupõe um modelo de qualidade a ser utilizado para a fase de classificação das anomalias.

Para o teste de classificação, submeteu-se o conjunto de alertas descrito na Seção 5.2.3 ao modelo obtido pelo treinamento descrito acima. A avaliação dos resultados consistiu na checagem da qualidade da classificação de cada uma das classes de eventos consideradas, além da *Acurácia Global*. Para tal, se realizou para cada classe o cálculo das métricas *TVP*, expressa pela Equação 2.2 e *Precisão*, expressa pela Equação 2.3. Para este fim, considerou-se cada conjunto de grupos representativos de um dado evento

de forma unificada. Para a medição do desempenho geral da classificação utilizou-se da *Acurácia Global*. Os resultados obtidos são apresentados na Figura 5-4.

Como se pode observar, o AutoClass com a abordagem CA atingiu uma *Acurácia Global* de 89,79%, sendo este valor fruto de um bom desempenho na classificação das anomalias verdadeiras e um desempenho razoável na classificação das falsas anomalias. O bom resultado na classificação das anomalias verdadeiras pode ser conferido pelo alto valor de *TVP* verificado nas classes representando os ataques. O valor de 100% de *Precisão* observada na classe de falsas anomalias também indica que nenhum alerta verdadeiro fora classificado nesta classe. Este é o comportamento ideal, pois a classificação de uma anomalia legítima como falsa adulteraria uma detecção correta realizada pelo AD, o que diminuiria sua taxa de *TVP*.



	V. de Portas	V. de Vuln. Web	At. de Dicionário	V. de Rede	DoS	FA
TVP	100%	99,70%	94,24%	100%	99,46%	64,47%
Precisão	99,42%	65,24%	100%	99,44%	95,62%	100%

Figura 5-4. Qualidade geral da classificação - abordagem CA.

A *TVP* de 64,47% verificada para a classe de falsas anomalias corresponde a porcentagem dos falsos positivos que foram corretamente identificados. Por mais que este seja um valor somente moderado no contexto dos sistemas de classificação, isso indica que dos 6.330 falsos positivos detectados pelo AD, 4.081 foram corretamente identificados e poderiam ser

descartados das análises de um analista, poupando-lhe um tempo considerável.

A maioria dos falsos positivos que não foram corretamente classificados nas classes correspondentes as falsas anomalias, foram classificados junto às classes dos ataques de varredura de vulnerabilidades *web.* Isso justifica valor reduzido de *Precisão* para as classes deste ataque em comparação às outras classes. Este comportamento se deu pelas caraterísticas deste ataque - várias requisições a partir de uma origem a um certo servidor *web* - se assemelhar a determinados comportamentos legítimos.

5.3.2 Resultados com a abordagem CFA

Os testes do AutoClass com a abordagem CFA se assemelharam com os realizados para a abordagem CA, diferenciando-se somente pelo conjunto de treinamento utilizado. Nesta abordagem, além dos dados trafegados entre as redes atacante e alvo, utilizou-se do tráfego legítimo gerado por usuários reais da rede interna do laboratório ACME!.

Com um aumento considerável da quantidade e da heterogeneidade dos dados de treinamento, este processo levou aproximadamente 45 minutos, o que representa um aumento de cerca de 51 vezes em comparação com o tempo obtido com a abordagem CA. Embora se tenha observado um aumento bastante considerável, o tempo de 45 minutos na fase de treinamento é ainda factível, uma vez que esta etapa deve ser realizada esporadicamente em ambientes reais. Com relação à quantidade de classes, nesta abordagem o AutoClass encontrou 34. Na Figura 5-5 pode ser visualizada a disposição dos dados de treinamento em cada uma das classes.

Observa-se que mesmo com o aumento considerável na heterogeneidade do conjunto de treinamento, o AutoClass conseguiu gerar classes com um alto grau de HIC. Esta qualidade se refletiu na HGP, que neste experimento foi de 99,57%, um valor superior àquele alcançado com a abordagem CA. Este alto valor alcançado pressupõe que este modelo de classes será eficaz na classificação dos alertas.

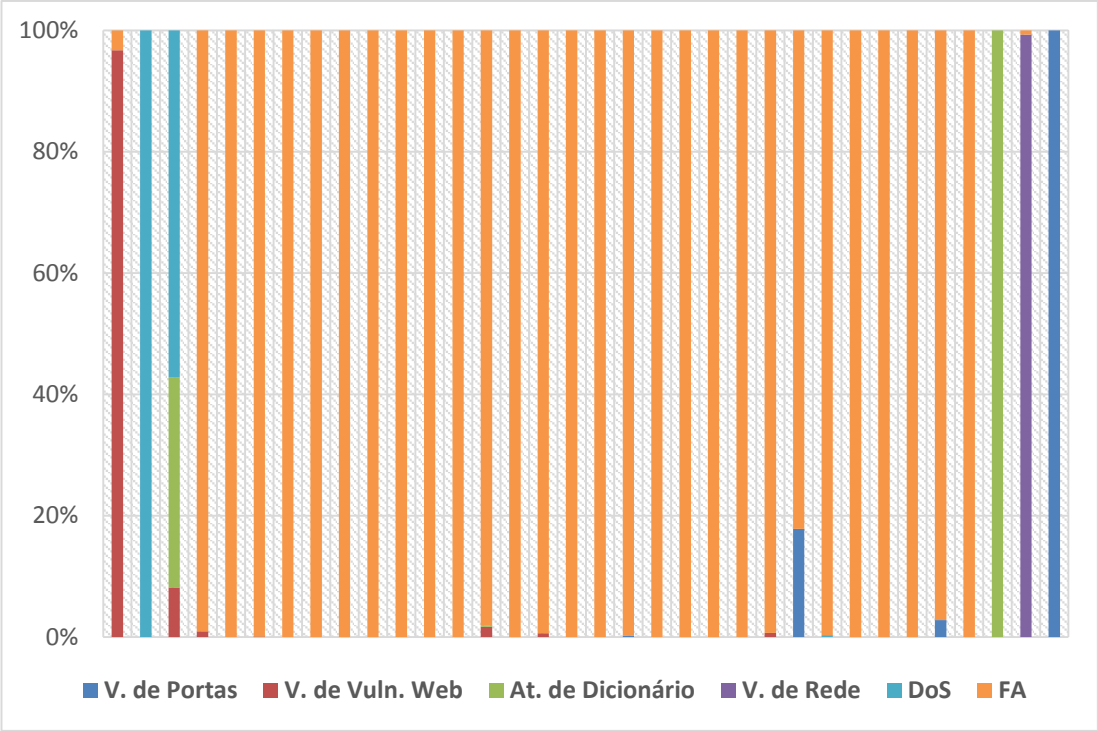
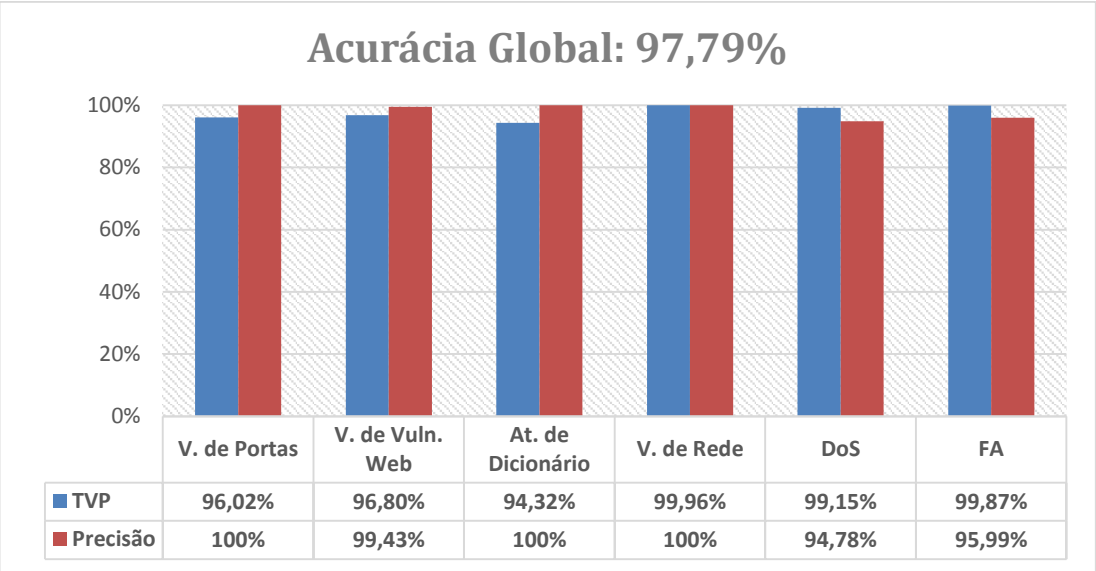


Figura 5-5. Disposição das instâncias nas classes de treinamento.

Com relação ao mapeamento das classes, verifica-se que as instâncias referentes ao ataque de varredura de portas foram majoritariamente agrupadas por uma classe, as do ataque de varredura de vulnerabilidades *web* por uma, as do ataque de dicionário em uma classe, as do ataque de varredura de rede em uma classe, as do ataque de DoS em duas classes e as instâncias referentes ao tráfego legítimo foram majoritariamente agrupadas por 28 classes.

Os testes da classificação foram realizados nos mesmos moldes discutidos na abordagem CA. Na Figura 5-6 podem ser vistos a *TVP* e a *Precisão* para cada classe, assim como a *Acurácia Global* obtida. Neste experimento observou-se uma *Acurácia Global* de 97,79%, superior àquela obtida na abordagem CA. Essa melhoria é explicada pela adição das instâncias de treinamento correspondentes ao tráfego da rede do Laboratório ACME!. Com a adição dessas novas informações, o AutoClass construiu um modelo mais preciso para o reconhecimento do tráfego legítimo. A eficácia deste novo modelo pode ser comprovada pelo aumento de mais de 35 pontos percentuais no novo valor da *TVP* verificado para as classes de falsas

anomalias. O valor de 99,87% na TVP implica que dos 6.330 falsos positivos detectados pelo AD, 6.322 foram identificados, o que representa praticamente todo o conjunto de falsos positivos.



	V. de Portas	V. de Vuln. Web	At. de Dicionário	V. de Rede	DoS	FA
TVP	96,02%	96,80%	94,32%	99,96%	99,15%	99,87%
Precisão	100%	99,43%	100%	100%	94,78%	95,99%

Figura 5-6. Qualidade geral da classificação - abordagem CFA.

Entretanto, essa melhoria vem a um custo. Como era esperado, de acordo com a discussão na Seção 5.2.2, com o grande aumento no número de classes encontradas na etapa de treinamento, a classificação tonou-se mais complexa. Em decorrência disso, houve um leve decréscimo no desempenho da classificação das verdadeiras anomalias. Na Figura 5-6, observa-se uma redução da *Precisão* das falsas anomalias em comparação com a abordagem CA. O novo valor de aproximadamente 96% indica que 4% dos alertas verdadeiros foram classificados como falsas anomalias, o que causou adulteração em algumas decisões corretas tomadas pelo AD.

5.4 Resultados com o algoritmo OPFC

Nesta seção são descritos os resultados com o algoritmo OPFC, explicado na Seção 2.5.1. Os testes com este algoritmo seguiram a mesma metodologia verificada nos testes com o AutoClass, assim alguns pontos já

bem detalhados na Seção 5.3 serão subentendidos para os resultados do OPFC.

Como o algoritmo OPFC faz uso da distância Euclidiana para a construção de seu modelo de treinamento, optou-se pela normalização dos dados como sugerido em (WITTEN; FRANK, 2000), já que características de maior escala poderiam exercer uma influência indevida no modelo em detrimento das características de menor escala. Para isso, se utilizou o logaritmo na base 2 das características *total_src_pkts* e *total_src_bytes*, uma vez que estas não apresentam um intervalo específico de valores, diferentemente de todas as outras características que são definidas no intervalo *[0,1]*. A normalização logarítmica é bastante comum e tem demonstrado bons resultados com os algoritmos de agrupamento [(ERMAN et al, 2006b), (PAXSON, 1994)]. Além do OPFC, esta mesma normalização foi realizada para os algoritmos K-médias e X-médias pelas mesmas considerações acima.

Outro aspecto que difere o OPFC do AutoClass é a necessidade de se determinar o melhor parâmetro k_{max} para os experimentos. Assim, além dos resultados das fases de treinamento e classificação, serão abordados os detalhes da escolha de k_{max} para as abordagens CA e CFA.

5.4.1 Resultados com a abordagem CA

Antes de proceder com a etapa de treinamento do algoritmo OPFC é preciso definir o melhor valor para o parâmetro k_{max}. Neste trabalho, este valor foi determinado por experimentação, assim como realizado em outros trabalhos que se utilizaram do OPFC para outros fins [(ROCHA et al, 2009), (MARTINS et al, 2014)]. Para isso, testou-se o valor de k_{max} no intervalo [5,105] com incrementos de 10. Para cada teste, avaliou-se a HGP obtida e também a quantidade de classes encontradas. Além de um valor elevado para HGP, que representa a qualidade do modelo encontrado em termos de pureza, busca-se o menor número possível de classes, já que um número elevado prejudica de forma geral a performance do classificador em termos de memória e processamento. Entretanto, o que geralmente se nota é uma

relação inversa entre essas duas medidas, pois classificadores com maior número de classes tendem a demonstrar melhores resultados quanto à *Acurácia Global*. Os resultados obtidos na experimentação dos valores para k_{max} são apresentados na Figura 5-7.

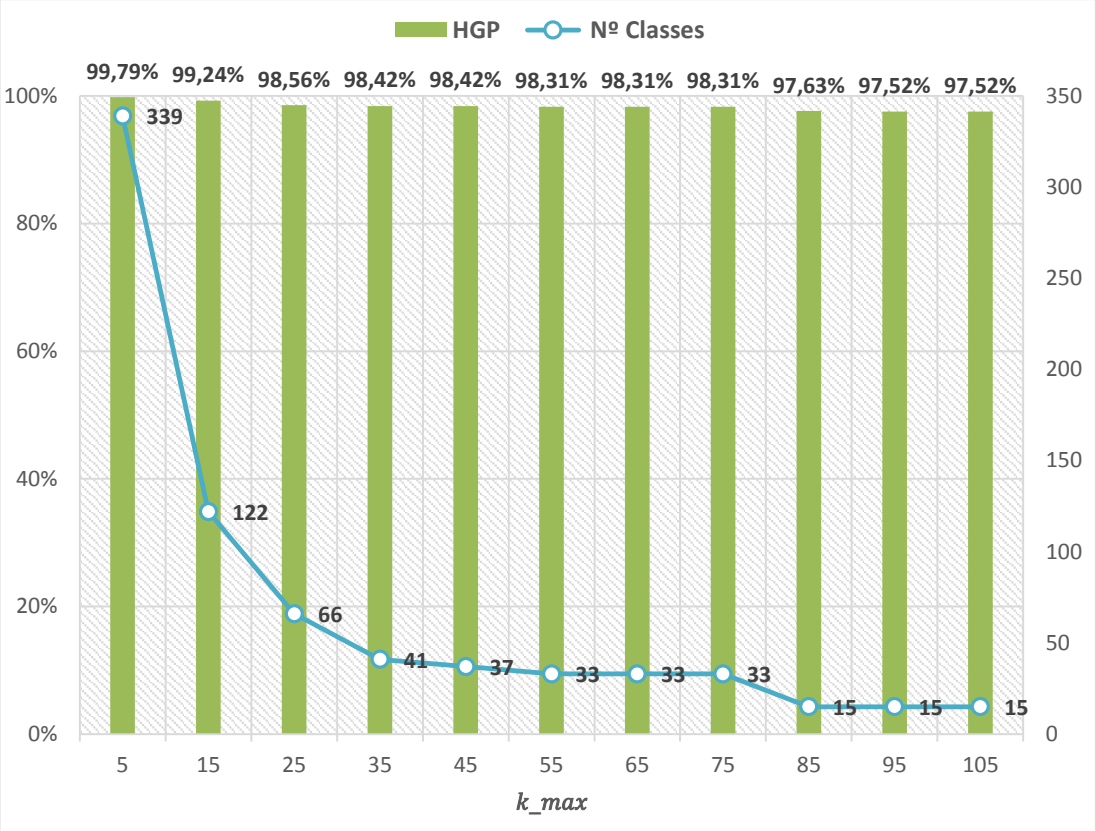


Figura 5-7. Experimentação com valores para k_{max} – abordagem CA.

Como pode ser visto, o valor de k_{max} tem bastante influência na quantidade de classes encontradas durante o treinamento. Como explicado na Seção 2.5.1, o valor de k_{max} impacta diretamente na definição de *k*, utilizado para a construção da relação de adjacência A_k. No OPFC, quanto menor o valor de k menor é a quantidade de vizinhos diretos de uma dada instância s. Com regiões de influência menores, mais instâncias se despontam como protótipos em suas regiões e, portanto, a quantidade de classes é maior em comparação a modelos com valores superiores de k. Este comportamento se confirmou nos testes realizados, podendo-se observar que quanto maior o valor de k_{max} menor é a quantidade de classes encontradas. Uma

consequência direta da redução na quantidade de classes é um decréscimo no valor de HGP obtido para cada teste.

Para a definição de k_{max} procurou-se, inicialmente, um balanço entre os valores de HGP e a quantidade de classes encontradas. Entretanto, o sistema de classificação se mostrou altamente sensível aos valores de HGP no uso do algoritmo OPFC, implicando que pequenos decréscimos no valor de HGP resultassem em altas perdas da *Acurácia Global* na etapa posterior de classificação. Dessa forma, deu-se um peso maior ao valor de HGP em detrimento do número de classes. Analisando-se o gráfico na Figura 5-7, constata-se uma grande redução no número de classes quando se eleva o valor de k_{max} de 5 para 15, em contrapartida, vê-se uma redução ínfima no valor de HGP. Ao se elevar k_{max} para 25, constata-se uma redução na proporção entre os decréscimos no número de classes e os valores de HGP, sendo que o valor de HGP teve uma queda levemente maior e a quantidade de classes não diminuiu tão consideravelmente como ocorreu para k_{max} igual a 15. Por este motivo escolheu-se 15 para o valor de k_{max} no treinamento com a abordagem CA.

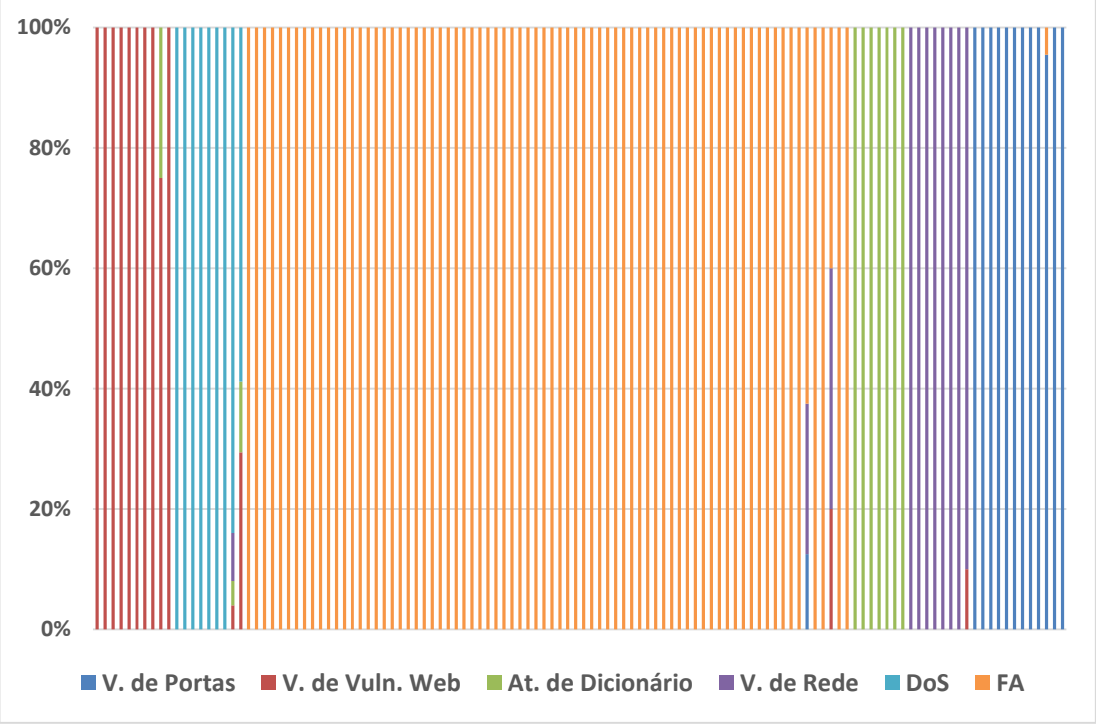
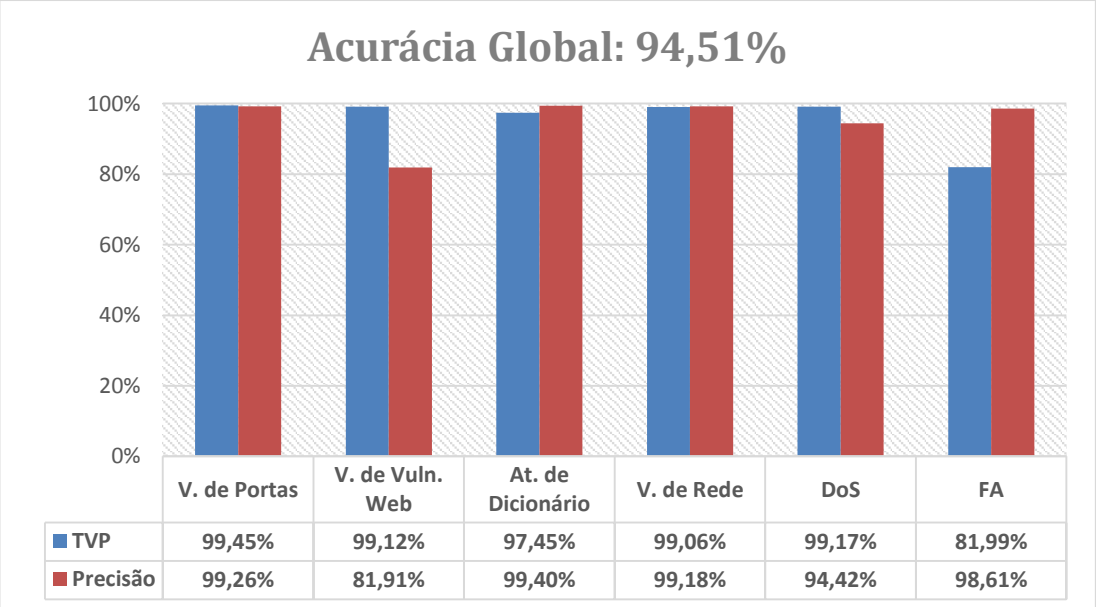


Figura 5-8. Disposição das instâncias nas classes de treinamento.

Com essa configuração, a etapa de treinamento com os dados da abordagem CA levou cerca de 1,3 segundos e foram encontradas 122 classes. Na Figura 5-8 pode ser vista a disposição das instâncias de treinamento entre as classes geradas.

Ao se analisar as cores no gráfico ilustrado na Figura 5-8 percebe-se um bom nível de homogeneidade entre as classes, o que é confirmado por uma HGP de 99,24% para este experimento. Apesar de um considerável aumento na quantidade de classes em comparação com os experimentos juntos ao AutoClass, obteve-se uma HGP superior, o que refletiu na qualidade da classificação realizada.

Para os testes de classificação com o conjunto de anomalias obteve-se os resultados exibidos na Figura 5-9. A *Acurácia Global* obtida pelo OPFC foi de 94,51%, superior a acurácia obtida pelo AutoClass, como foi previsto pelo valor superior de HGP obtido. A maior diferença entre os dois algoritmos foi com relação aos falsos positivos relativos às classes representativas dos ataques de varredura de vulnerabilidades *web*, por mais que os algoritmos tendam a confundir algumas falsas anomalias com este tipo de ataque, o OPFC obteve uma *Precisão* superior em mais de 15 pontos percentuais para esta classe em relação ao AutoClass.



	V. de Portas	V. de Vuln. Web	At. de Dicionário	V. de Rede	DoS	FA
TVP	99,45%	99,12%	97,45%	99,06%	99,17%	81,99%
Precisão	99,26%	81,91%	99,40%	99,18%	94,42%	98,61%

Figura 5-9. Qualidade geral da classificação - abordagem CA.

Em contrapartida, observa-se que o AutoClass obteve 100% de *Precisão* para as classes de falsas anomalias, indicando que este foi superior em separar os alertas de anomalias verdadeiras das falsas anomalias. Já o OPFC atingiu uma *Precisão* de 98,61% para a classe de falsas anomalias, o que indica que 73 alertas verdadeiros foram classificados junto às classes de falsas anomalias, o que adulterou algumas detecções corretas realizadas pelo AD.

5.4.2 Resultados com a abordagem CFA

Assim como feito com a abordagem CA, realizou-se os experimentos para se encontrar o melhor k_{max} para a abordagem CFA, dado que esta abordagem contempla uma adição bastante considerável de dados na fase de treinamento, o que poderia ter efeito sobre o valor de k_{max}. Para os testes utilizou-se a mesma metodologia descrita na abordagem CA e os resultados são resumidos na Figura 5-10.

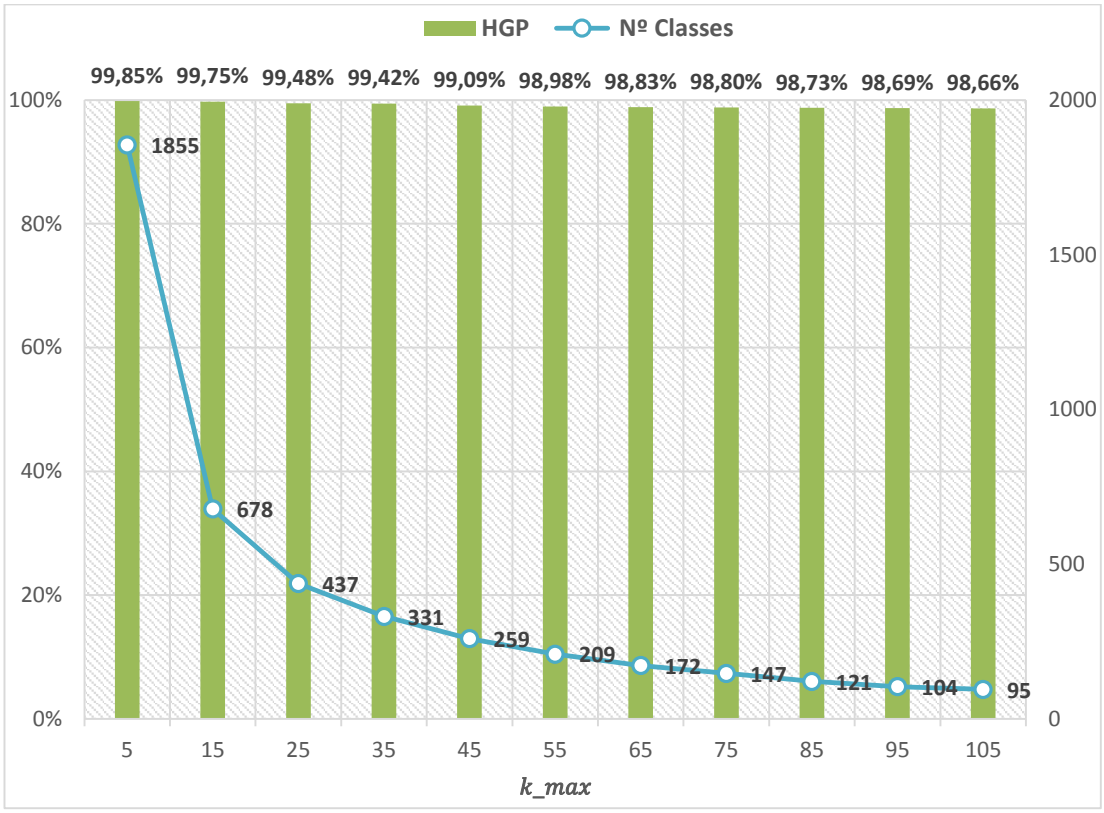


Figura 5-10. Experimentação com valores para k_{max} - abordagem CFA.

Nestes resultados é possível observar um comportamento bastante parecido com o obtido com a abordagem CA, com diferenças somente na quantidade de classes encontradas, o que é novamente explicado pela adição das instâncias representantes do tráfego legítimo obtido na rede do laboratório ACME!. Levando-se em conta as mesmas considerações feitas para a abordagem CA, também se escolheu 15 para o valor de k_{max} desta abordagem.

Com a definição do parâmetro k_{max}, partiu-se para a etapa de treinamento. Esta etapa levou cerca de 46 segundos e 678 classes foram encontradas. A disposição das instâncias entre as classes é mostrada na Figura 5-11. Mesmo com a grande quantidade de classes encontradas, uma análise da Figura 5-11 permite a verificação de uma alta homogeneidade entre elas, a HGP foi de 99,75%, superior àquela obtida pelo AutoClass na abordagem CFA e pelo próprio OPFC na abordagem CA.

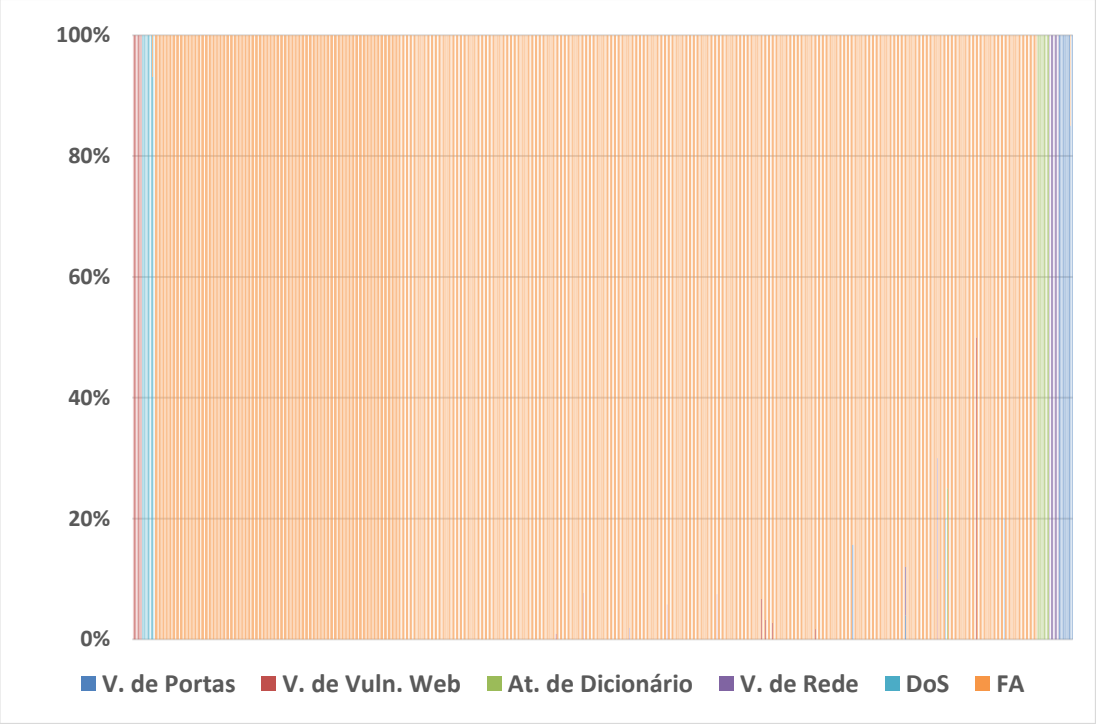
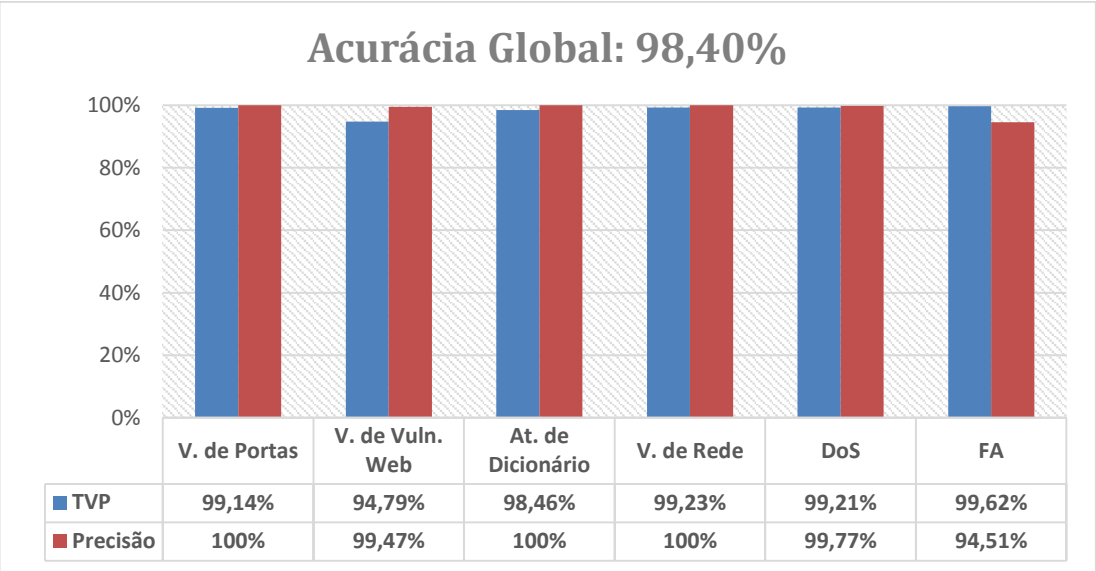


Figura 5-11. Disposição das instâncias nas classes de treinamento.

Os resultados do teste de classificação com o modelo obtido são apresentados na Figura 5-12. Neste experimento verificou-se uma *Acurácia*

Global de 98,40%, a qual foi obtida graças ao bom desempenho do OPFC em identificar os falsos positivos, aferido pela *TVP* de 99,62% da classe de falsas anomalias.



	V. de Portas	V. de Vuln. Web	At. de Dicionário	V. de Rede	DoS	FA
TVP	99,14%	94,79%	98,46%	99,23%	99,21%	99,62%
Precisão	100%	99,47%	100%	100%	99,77%	94,51%

Figura 5-12. Qualidade geral da classificação - abordagem CFA.

Isso indica que dos 6.330 falsos positivos detectados, somente 24 não foram identificados corretamente. No entanto, assim como nos experimentos com o AutoClass, essa melhoria veio a custo de um leve decréscimo na habilidade de se classificar os alertas verdadeiros. Na abordagem CFA constata-se um decréscimo de pouco mais de 4 pontos percentuais da *Precisão* da classe de falsas anomalias em comparação com a abordagem CA, significando que 366 alertas verdadeiros foram erroneamente classificados como falsas anomalias.

5.5 Resultados com o algoritmo K-médias

Nesta seção são apresentados os resultados do K-médias que é descrito na Seção 2.5.3. Novamente seguiu-se a metodologia verificada nos testes com o AutoClass e o OPFC. Assim como ocorreu para o OPFC, também foi

realizada a normalização dos dados, pois o K-médias se baseia fortemente no uso de distância Euclidiana para a criação de seu modelo de classificação.

Antes de se iniciar a etapa de treinamento é necessário encontrar o melhor valor para o parâmetro k, que representa a quantidade de classes desejada para o modelo de classificação. Alguns trabalhos [(MUDA et al, 2011), (LIN et al, 2014)] definem *k* de acordo com a quantidade de classes reais do problema tratado como se sempre houvesse uma correspondência de um para um entre as classes reais do problema e as classes encontradas pelo algoritmo de agrupamento. Esta é uma abordagem bastante simplista, pois como já visto, os algoritmos não supervisionados tendem a encontrar uma quantidade maior de classes do que a quantidade real de aplicações em consideração [(ZANDER et al, 2005), (NGUYEN; ARMITAGE, 2008)]. Por este motivo, esse trabalho contemplou testes com diferentes valores para *k* para a determinação dos melhores parâmetros para ambas as abordagens.

5.5.1 Resultados com a abordagem CA

O primeiro passo antes de se iniciar os testes de treinamento com o K-médias é a definição do melhor valor para *K*, considerando o conjunto de instâncias da abordagem CA. Com este fim, realizou-se experimentos com diferentes valores de *k* no intervalo $[6, 106]$. Escolheu-se o valor inicial 6 por ser esta a quantidade real de classes do problema, sendo que, dessa forma, o algoritmo não poderia encontrar um valor menor. A avaliação de cada teste foi realizada por meio da métrica HGP e novamente se procurou um balanço entre a quantidade de classes e o valor de HGP. Os resultados obtidos são resumidos na Figura 5-13.

A partir do gráfico exibido na Figura 5-13, se observa que com 56 classes uma HGP de 99,24% é obtida, além disso pode ser verificado que o valor de HGP tende a se estabilizar neste valor, mesmo com valores maiores para *k*. Dessa forma o parâmetro *k* foi definido como 56 para os testes com a abordagem CA.

Com essa configuração o treinamento do K-médias levou cerca de 1,2 segundos para o agrupamento das instâncias às 56 classes determinadas. A

disposição das instâncias nas classes pode ser vista na Figura 5-14. No gráfico é possível observar uma boa qualidade para o modelo obtido pelo K-médias, neste experimento a HGP foi de 99,24%. Os resultados da classificação das anomalias com este modelo são apresentados na Figura 5-15.

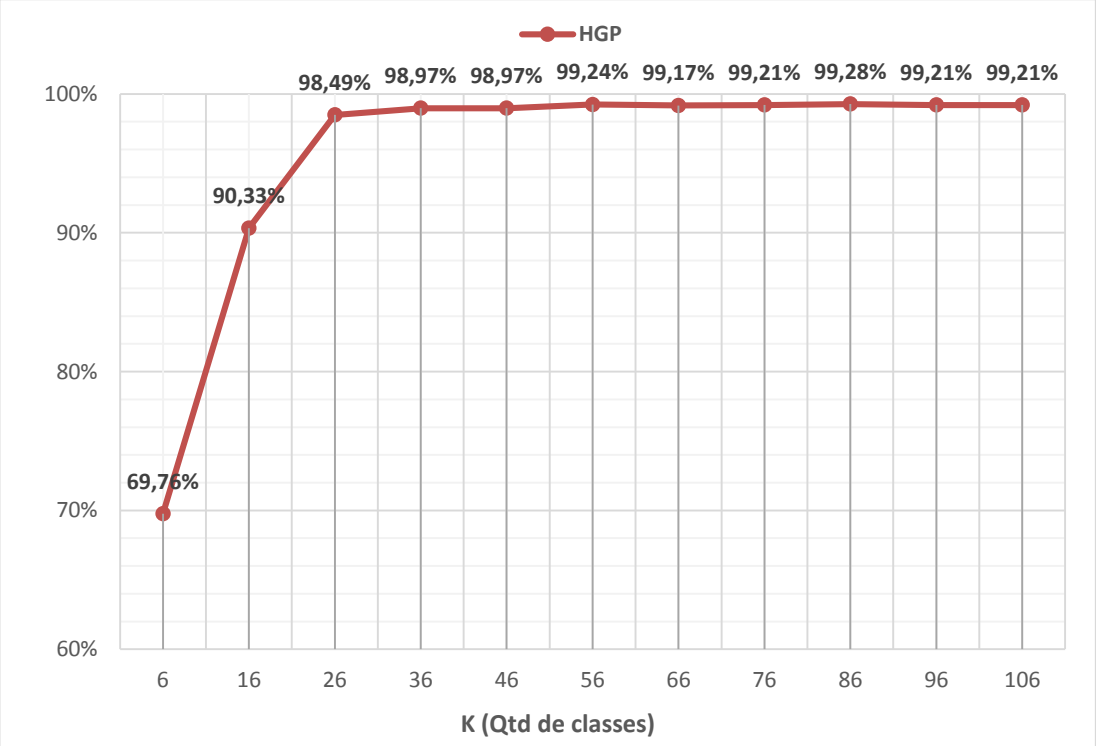


Figura 5-13. Experimentação com diferentes valores para K.

A *Acurácia Global* obtida neste experimento foi de 91,26%. Seguindo a tendência dos outros experimentos na abordagem CA, o K-médias se mostrou bastante capaz de classificar os alertas verdadeiros em suas classes corretas. Assim como com o AutoClass, obteve-se 100% de *Precisão* para a classe de falsas anomalias, indicando que nenhum alerta verdadeiro foi confundido como sendo uma falsa anomalia. No entanto, foi também demonstrada uma habilidade inferior para a correta classificação dos falsos positivos, seguindo a tendência de classificar uma parte deles junto às classes representantes da varredura de vulnerabilidades *web*.

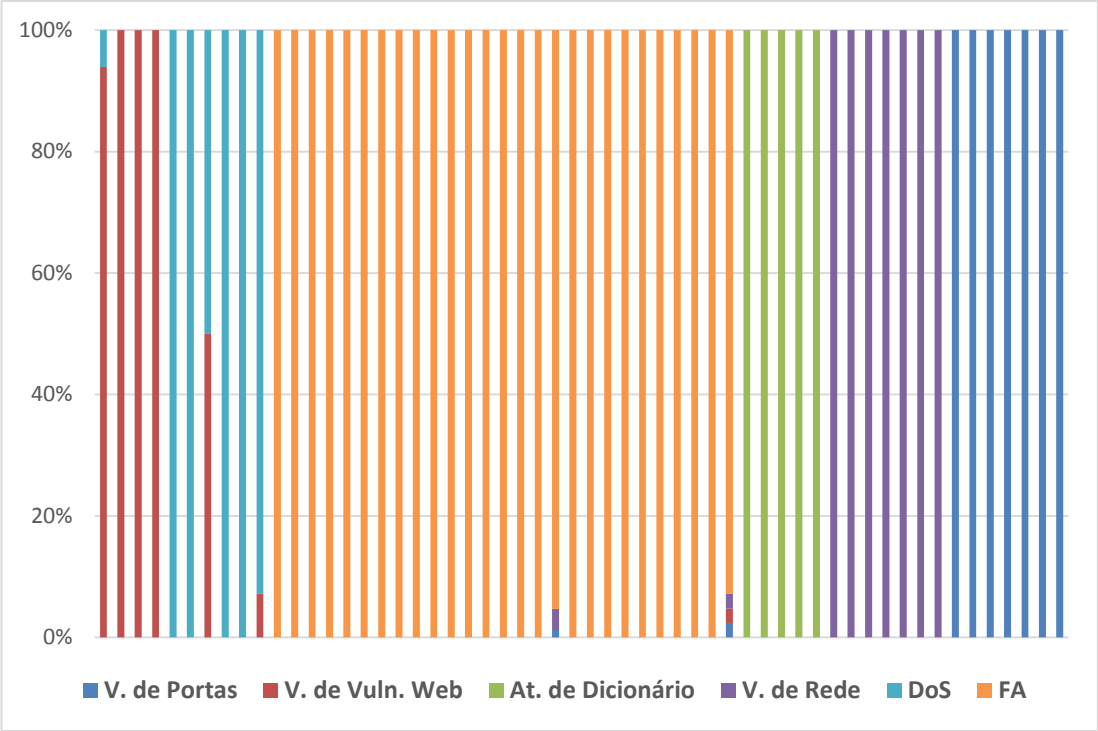
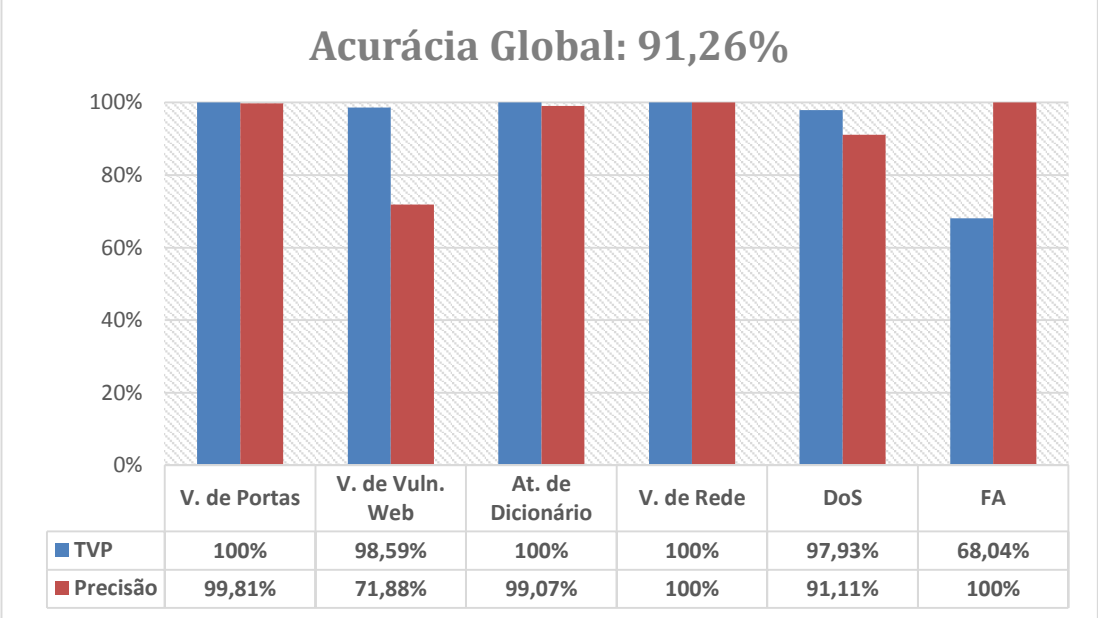


Figura 5-14. Disposição das instâncias nas classes de treinamento.



	V. de Portas	V. de Vuln. Web	At. de Dicionário	V. de Rede	DoS	FA
TVP	100%	98,59%	100%	100%	97,93%	68,04%
Precisão	99,81%	71,88%	99,07%	100%	91,11%	100%

Figura 5-15. Qualidade geral da classificação - abordagem CA.

5.5.2 Resultados com a abordagem CFA

Para a determinação do melhor parâmetro *k* para esta abordagem, foi seguida a mesma metodologia de experimentos da abordagem CA. Os

resultados obtidos são expostos na Figura 5-16. No gráfico exibido é mostrada uma oscilação crescente do valor da HGP conforme o valor de *k* vai sendo aumentado. Quando a quantidade de classes é elevada de 66 para 76, é visto um crescimento de 0,003% na HGP. Após este crescimento, nota-se uma estabilização no valor de HGP, pois ao se elevar a quantidade de 76 para 86 classes é percebido um crescimento de somente 0,0007% na HGP e de somente 0,0004% quando se eleva a quantidade de 96 para 106 classes. Sabe-se que com a aproximação de 100%, as elevações de HGP tendem se tornarem menores, entretanto, a estabilização notada foi suficiente para a nossa escolha devido a consideração do impacto acarretado pelo aumento na quantidade classes. Assim, escolheu-se 76 como valor para o parâmetro *k* no treinamento com os dados da abordagem CFA.

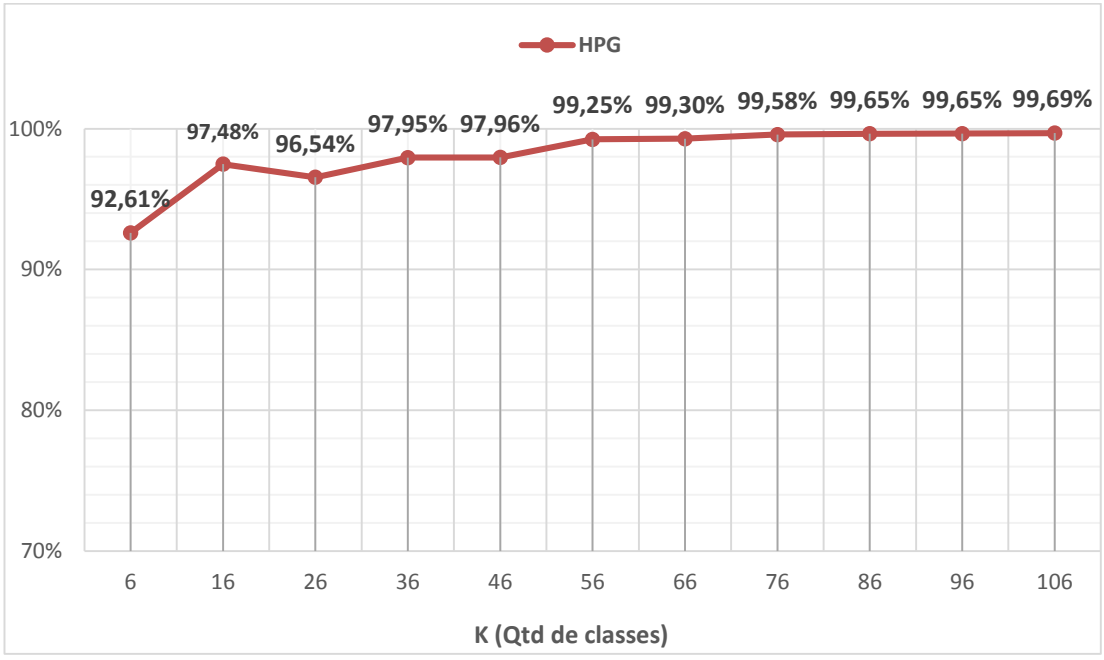


Figura 5-16. Experimentação com diferentes valores para K.

O treinamento do K-médias nessas configurações levou cerca de 3 segundos. A distribuição das instâncias entre as 76 classes é ilustrada na Figura 5-17.

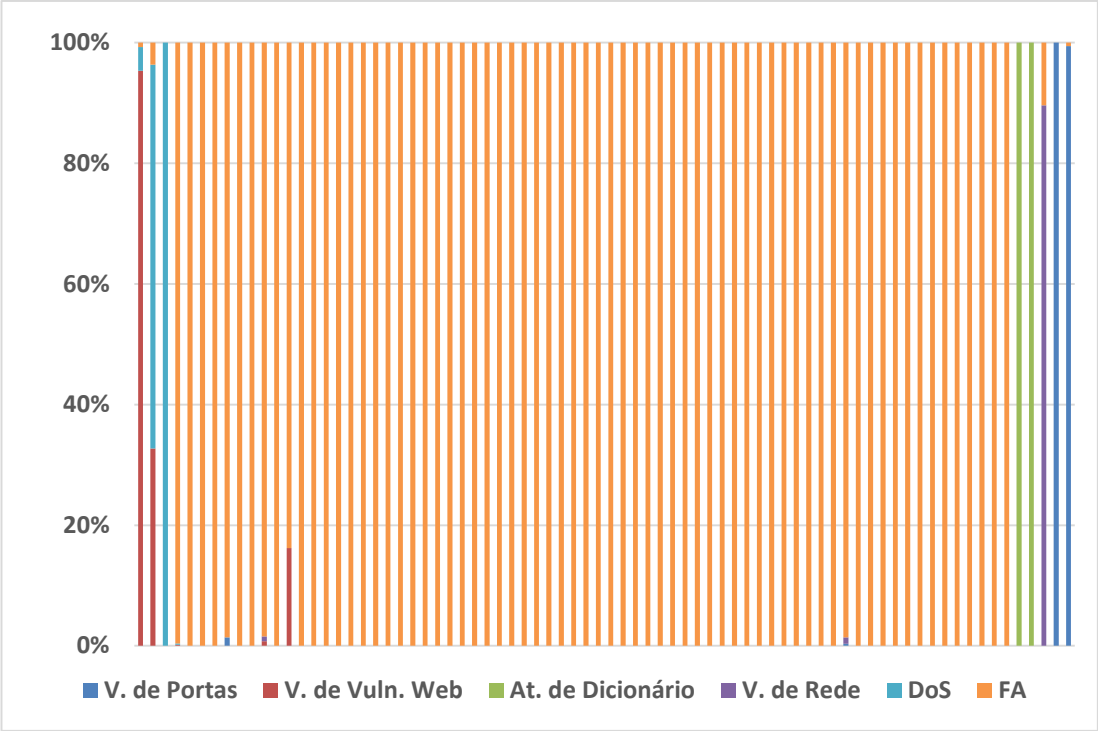
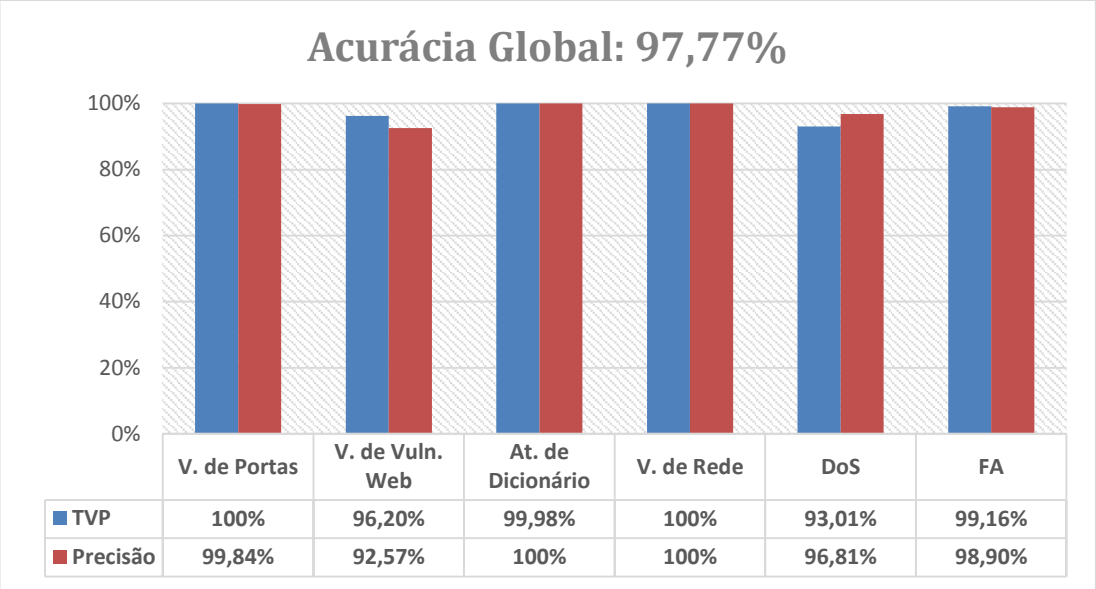


Figura 5-17. Disposição das instâncias nas classes de treinamento.

Como previsto nos experimentos com o valor *k*, o K-médias obteve uma HGP de 99,58%, um valor alto que indica um bom modelo para a classificação, os resultados são exibidos na Figura 5-18. Neste experimento foi verificada uma *Acurácia Global* de 97,77%, uma boa medida para sistemas de classificação no geral. No entanto, este foi o valor mais inferior entre todos os algoritmos testados com a abordagem CFA. Dentre todos os eventos, o K-Médias mostrou certa dificuldade em classificar os ataques de DoS em suas respectivas classes, classificando 302 destes alertas junto à classe dos ataques de varredura de vulnerabilidades *web*, o que impactou diretamente na *Precisão* desta última.

Apesar dos resultados negativos, o K-médias conquistou, simultaneamente, elevados valores para as métricas de *Precisão* e *TVP* para a classe de falsas anomalias. As métricas de *TVP* e *Precisão*, costumam ser inversamente proporcionais, o que torna difícil de se obter resultados elevados para essas duas métricas ao mesmo tempo (NGUYEN; ARMITAGE, 2008). Outro trunfo do K-médias diz respeito aos seus tempos de execução, sendo os menores dentre todos os algoritmos.



	V. de Portas	V. de Vuln. Web	At. de Dicionário	V. de Rede	DoS	FA
TVP	100%	96,20%	99,98%	100%	93,01%	99,16%
Precisão	99,84%	92,57%	100%	100%	96,81%	98,90%

Figura 5-18. Qualidade geral da classificação - abordagem CFA.

5.6 Resultados com o algoritmo X-médias

Os resultados obtidos na validação do X-médias são detalhados nesta seção. Este algoritmo, como explicado na Seção 2.5.4, consiste num aperfeiçoamento aplicado ao algoritmo K-médias. Sendo assim, suas características são bastante similares ao seu precursor. No entanto, importantes diferenças são notadas. Para os testes, a diferença mais importante é com relação à determinação do parâmetro *k*. Diferentemente do que ocorre com o K-médias, o X-médias não exige a especificação explicita deste parâmetro. O que este faz é encontrar o melhor valor dentro de um intervalo $[K_{min}, K_{max}]$. Para a definição de K_{min} utilizou-se da presunção de que o modelo deve contemplar pelo menos 6 classes para a classificação dos eventos considerados. Já o valor para K_{max} foi definido de acordo com o melhor K encontrado para o algoritmo K-médias, assim sendo 56 para a abordagem CA e 76 para a abordagem CFA. Para a normalização dos dados se utilizou da mesma abordagem para os algoritmos OPFC e K-médias.

5.6.1 Resultados com a abordagem CA

No treinamento com os dados da abordagem CA, o X-médias definiu 56 como o melhor valor para *K*. O processo total de treinamento levou 1,5 segundos e a disposição das instâncias entre as classes geradas é representada na Figura 5-19.

Neste experimento o X-médias obteve uma HGP de 99,17%. Este modelo aplicado às anomalias do conjunto de classificação rendeu os resultados apresentados na Figura 5-20. Nela, nota-se que a *Acurácia Global* obtida foi de 90,77%. Destaca-se o valor de 100% obtido na taxa de *Precisão* na classe das Falsas Anomalias, o que indica que nenhum ataque, corretamente detectado, foi classificado como sendo um comportamento legítimo. As altas taxas de *TVP* verificadas nas classes representantes dos ataques revelam um alto desempenho na classificação das anomalias verdadeiras, já a baixa *TVP* da classe de Falsas anomalias indicam a tendência já verificada com a abordagem CA nos outros experimentos.

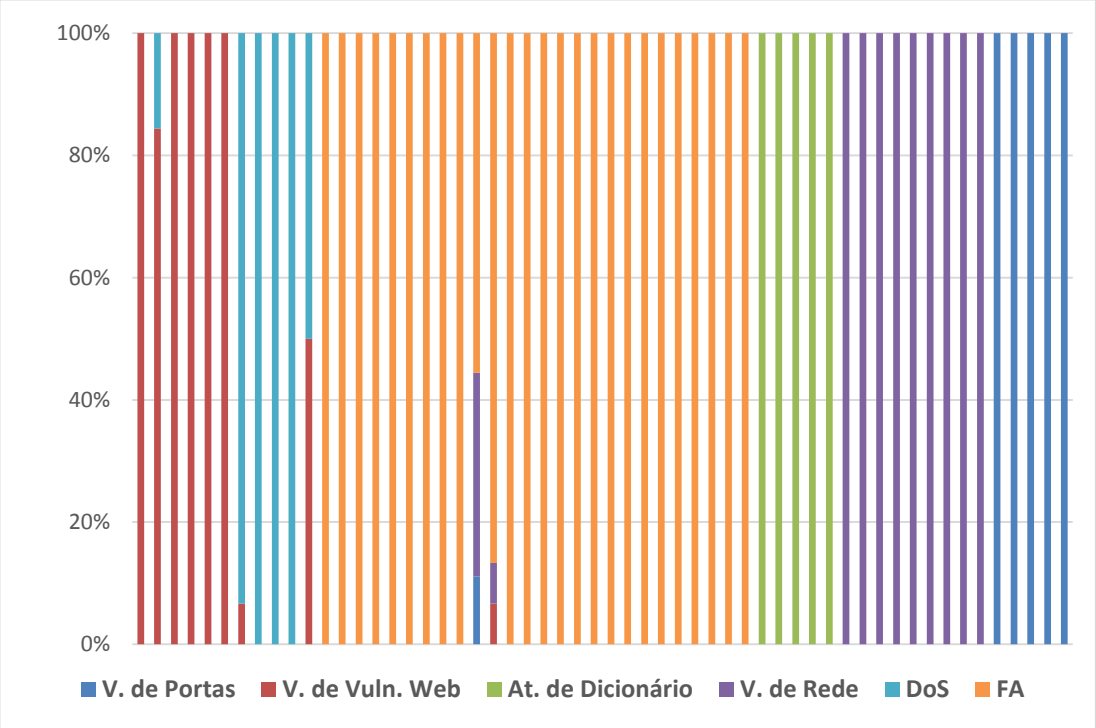
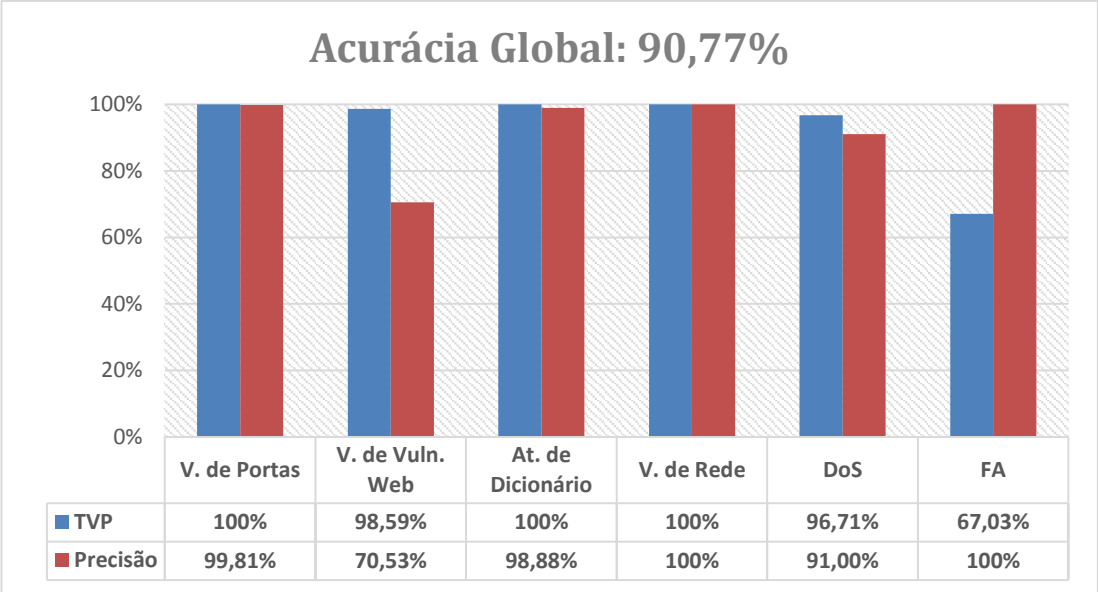


Figura 5-19. Disposição das instâncias nas classes de treinamento.



	V. de Portas	V. de Vuln. Web	At. de Dicionário	V. de Rede	DoS	FA
TVP	100%	98,59%	100%	100%	96,71%	67,03%
Precisão	99,81%	70,53%	98,88%	100%	91,00%	100%

Figura 5-20. Qualidade geral da classificação - abordagem CFA.

5.6.2 Resultados com a abordagem CFA

Semelhantemente ao ocorrido na abordagem CA, o X-médias também escolheu como valor para *k* o máximo valor K_{max} do intervalo fornecido, sendo neste caso 76. Nesta condição, fora gerado um modelo com uma HGP de 99,57%, bastante semelhante ao valor obtido pelo K-médias. A disposição das instâncias entre as classes geradas pode ser vista na Figura 5-21. Com a aplicação deste modelo para a classificação das anomalias, obteve-se uma *Acurácia Global* de 98,67%, os resultados relativos aos valores de *TVP* e *Precisão* são mostrados na Figura 5-22.

Com a abordagem CFA, o X-médias registrou o maior valor de *Acurácia Global* dentre todos os algoritmos, indicando, em termos gerais, que este foi o algoritmo com maior performance na classificação das anomalias. É possível notar altos valores para *TVP* e *Precisão* em todas as classes, com pequenas exceções. Como algumas anomalias do ataque de DoS foram erroneamente classificadas como varreduras de vulnerabilidade web, houve uma pequena redução nas taxas de *TVP* e *Precisão* destas, respectivamente. A *Precisão* de 99,27% para as classes de falsas anomalias também foi a maior entre todos os algoritmos na abordagem CFA e indica um alto potencial na separação entre as anomalias verdadeiras e falsas, mesmo com o grande aumento na heterogeneidade no conjunto de treinamento da abordagem CFA.

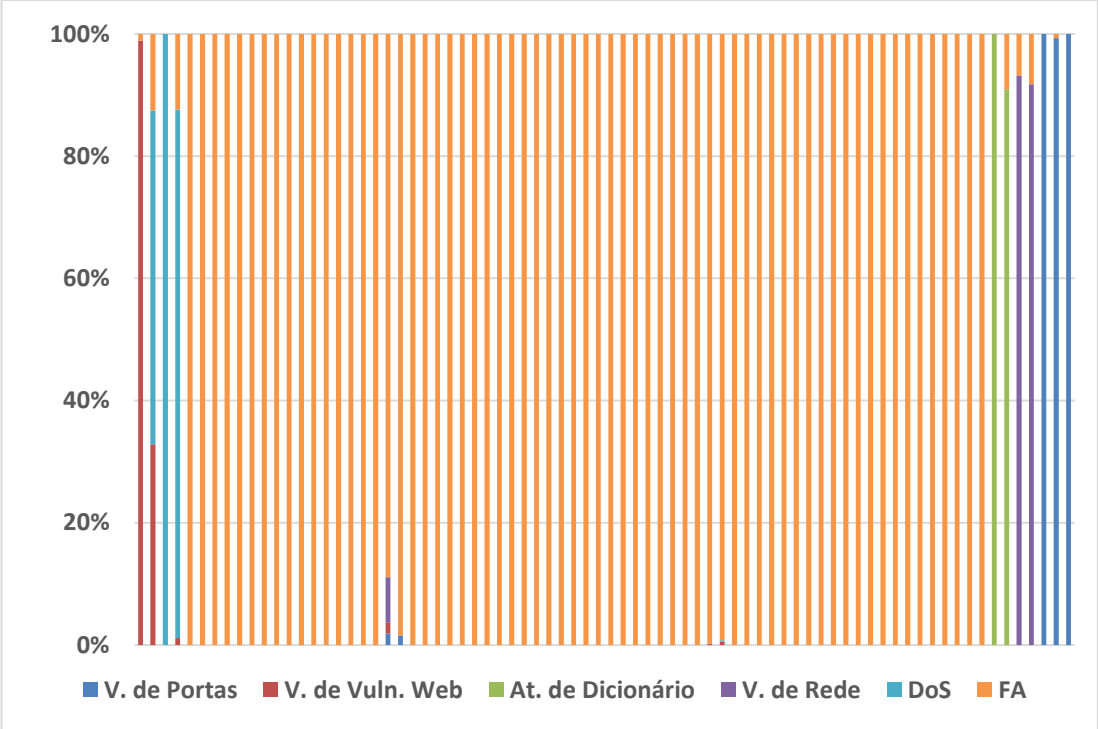
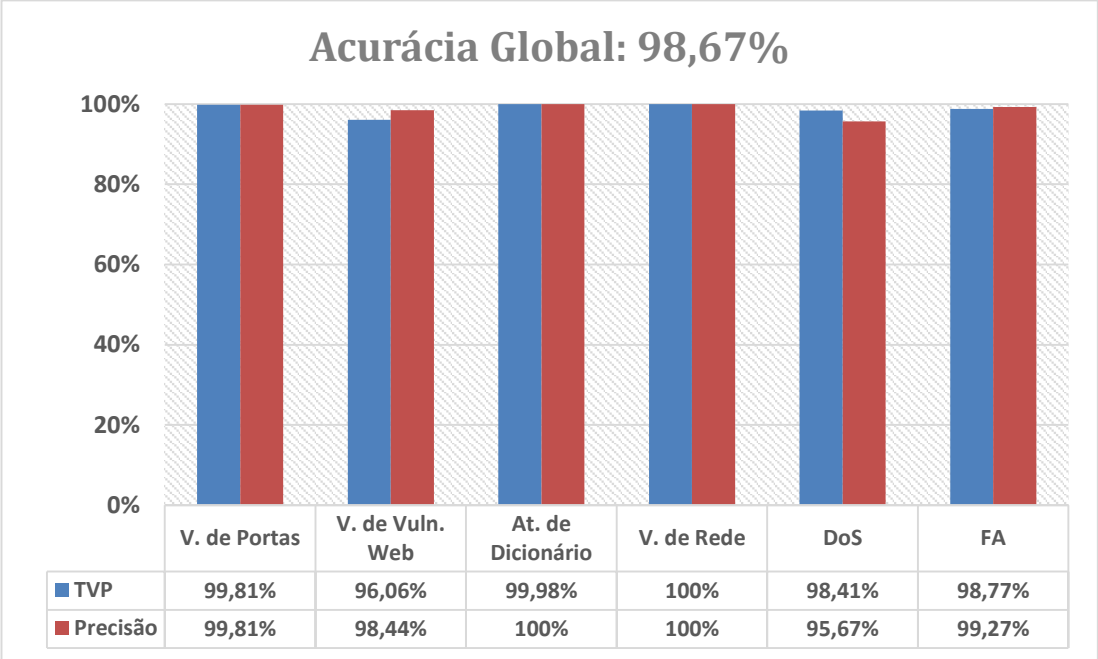


Figura 5-21. Disposição das instâncias nas classes de treinamento.



	V. de Portas	V. de Vuln. Web	At. de Dicionário	V. de Rede	DoS	FA
TVP	99,81%	96,06%	99,98%	100%	98,41%	98,77%
Precisão	99,81%	98,44%	100%	100%	95,67%	99,27%

Figura 5-22. Qualidade geral da classificação - abordagem CFA.

Com a análise dos resultados obtidos por todos os algoritmos experimentados, é possível verificar a superioridade da abordagem CFA

sobra a CA em termos de *Acurácia Global*. A premissa inicial era a obtenção de uma relação de complementaridade entre as duas abordagens, ou seja, as vantagens de uma abordagem se evidenciariam ante as desvantagens apresentadas pela outra. Entretanto, por mais que esta relação possa ser levemente notada, já que a abordagem CA se mostrou mais eficaz na classificação das anomalias verdadeiras, com a abordagem CFA classificou-se corretamente quase que a totalidade das falsas anomalias ao preço de uma leve redução no desempenho da classificação das verdadeiras anomalias. Assim conclui-se a superioridade da abordagem CFA, mesmo se considerando a pequena vantagem da abordagem CA na classificação dos verdadeiros positivos.

5.7 Comparação entre os algoritmos

Nesta seção é apresentada uma comparação entre o desempenho obtido pelos diferentes algoritmos em termos das métricas de *TVP*, *Precisão* e Medida-F, além dos tempos registrados nas etapas de treinamento e classificação. Os dados apresentados são aqueles obtidos com o uso da abordagem CFA, visto que com esta abordagem os algoritmos apresentaram os melhores resultados.

Na Figura 5-23 são apresentados os valores de *TVP* obtidos por cada algoritmo de acordo com cada ataque considerado, além das falsas anomalias. Um alto valor de *TVP* para uma certa classe indica sua habilidade em classificar corretamente as anomalias pertencentes a esta classe. Como se percebe pelo gráfico na Figura 5-23, o X-médias apresentou os melhores resultados com relação à *TVP*, com uma média de 98,84% contra 98,41% do OPFC, 98,06% do K-médias e 97,69% do AutoClass. É também possível notar que, no geral, todos os algoritmos tiveram certa dificuldade em classificar as anomalias relativas ao ataque de varredura de vulnerabilidades Web, o que se reflete pela *TVP* inferior desta classe. De fato, este foi o ataque que demonstrou o comportamento menos anômalo em comparação ao tráfego legítimo, por isso algumas anomalias deste evento foram classificadas como

falsas anomalias, outra parte foi classificada junto aos ataques de negação de serviço.

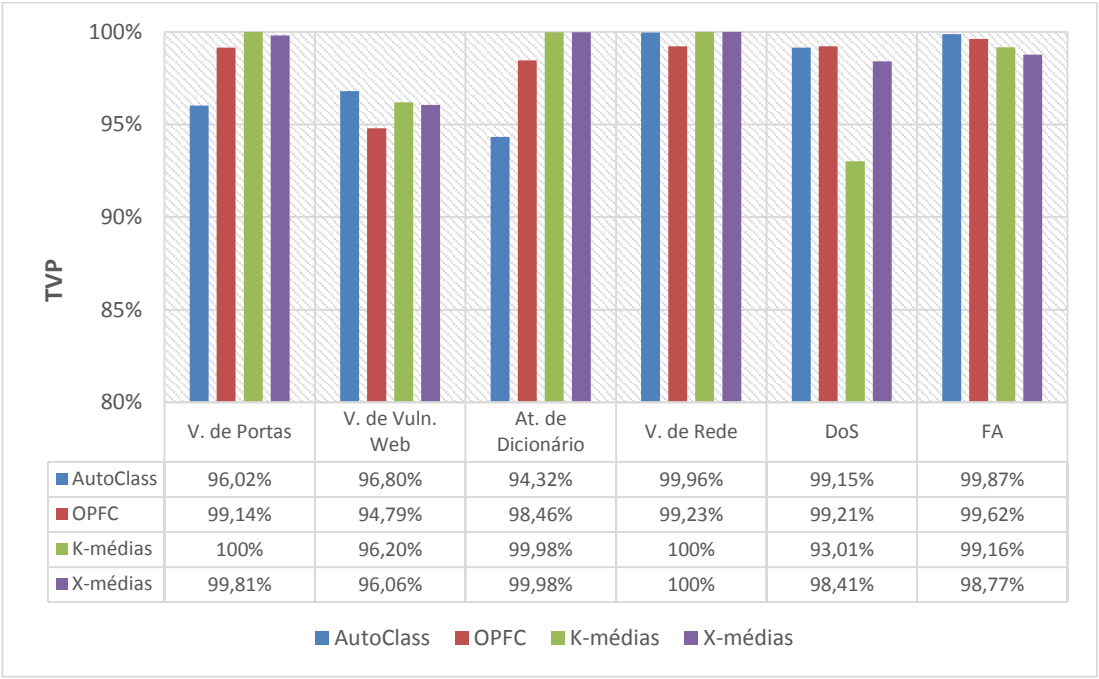
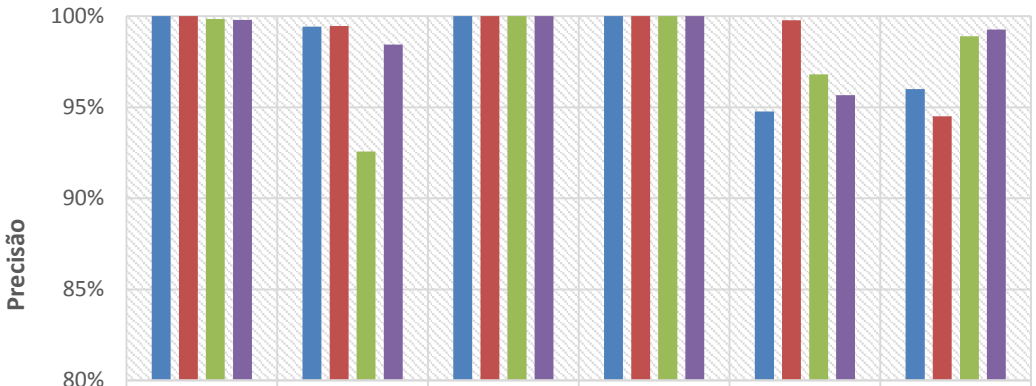


Figura 5-23. Valores de *TVP* obtidos por cada algoritmo avaliado.

Outro comportamento interessante é com relação a *TVP* obtida pelo AutoClass para a classe de falsas anomalias, a qual foi superior entre todos os outros algoritmos. Isso indica que o AutoClass foi o algoritmo mais eficiente na identificação dos falsos positivos detectados pelo AD, seguido pelo OPFC, depois pelo K-médias e por último o X-médias.

Na Figura 5-24 são exibidos os resultados relativos à *Precisão* obtida por cada algoritmo para cada um dos eventos considerados. A *Precisão* é a métrica responsável por indicar a quantidade de falsos positivos em cada classe. Ou seja, os eventos não pertencentes a uma certa classe X mas erroneamente classificados em X. Neste mérito, o algoritmo que mais se destacou foi o OPFC, com uma *Precisão* média de 98,96% contra 98,87% do X-médias, 98,37% do AutoClass e 98,02% do K-médias.



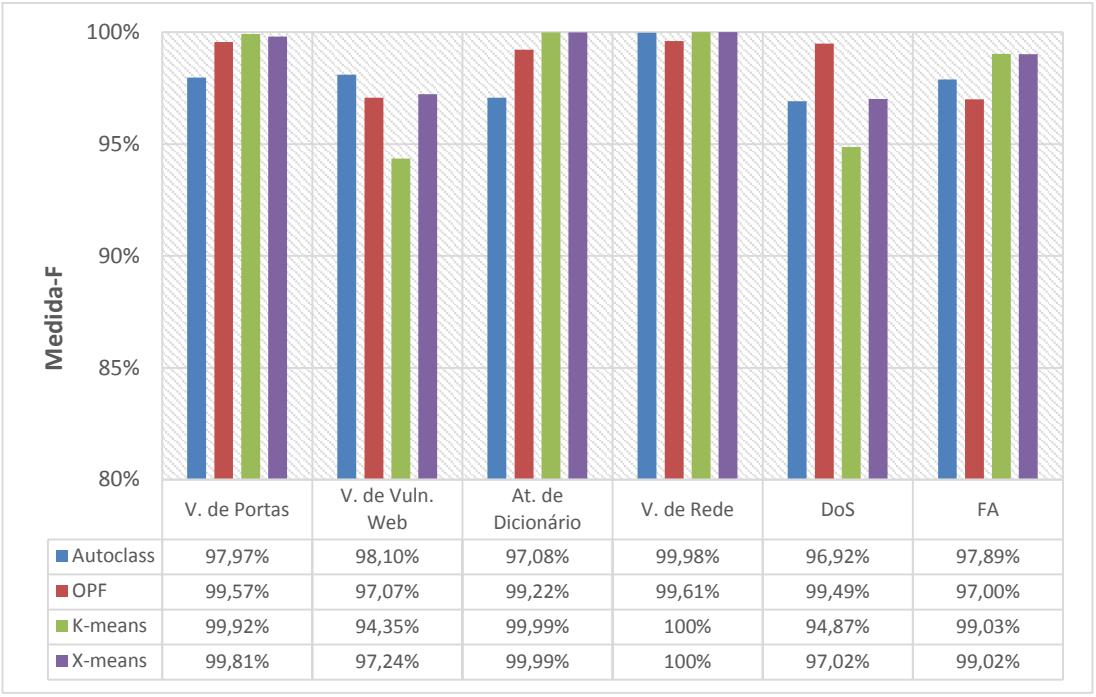
	V. de Portas	V. de Vuln. Web	At. de Dicionário	V. de Rede	DoS	FA
AutoClass	100%	99,43%	100%	100%	94,78%	95,99%
OPFC	100%	99,47%	100%	100%	99,77%	94,51%
K-médias	99,84%	92,57%	100%	100%	96,81%	98,90%
X-médias	99,81%	98,44%	100%	100%	95,67%	99,27%

Figura 5-24. Valores de *Precisão* obtidos por cada algoritmo avaliado.

Os valores inferiores obtidos pela maioria dos algoritmos nas classes do ataque de DoS e Falsas Anomalias se justificam, principalmente, pela classificação errada de algumas instâncias do ataque de varredura de vulnerabilidades web, como já explicado na avaliação da métrica *TVP*.

Quanto menor o valor da taxa de *Precisão* para a classe de verdadeiras anomalias, maior a quantidade de ataques corretamente detectados que foram erroneamente classificados como falsas anomalias. O impacto desta classificação errônea sobre a *Precisão* inicial obtida pelo AD é discutido na próxima seção.

A fim de se avaliar o balanço entre a *TVP* e a Precisão de cada algoritmo, foi utilizada a métrica Medida-F, que computa essas duas métricas de forma unificada. Os valores obtidos são apresentados na Figura 5-25. Com esta métrica é possível observar que o X-médias demonstrou o maior equilíbrio entre os casos de FPs e FNs, com uma Medida-F média de 98,85% contra 98,66% do OPFC, 98,03% do K-médias e 97,99% do AutoClass. Esse é um resultado importante, dado que geralmente se nota uma relação inversa entre os FPs e FNs e é preciso a minimização da ocorrência de ambos os casos.



	V. de Portas	V. de Vuln. Web	At. de Dicionário	V. de Rede	DoS	FA
Autoclass	97,97%	98,10%	97,08%	99,98%	96,92%	97,89%
OPF	99,57%	97,07%	99,22%	99,61%	99,49%	97,00%
K-means	99,92%	94,35%	99,99%	100%	94,87%	99,03%
X-means	99,81%	97,24%	99,99%	100%	97,02%	99,02%

Figura 5-25. Valores de Medida-F para os algoritmos avaliados.

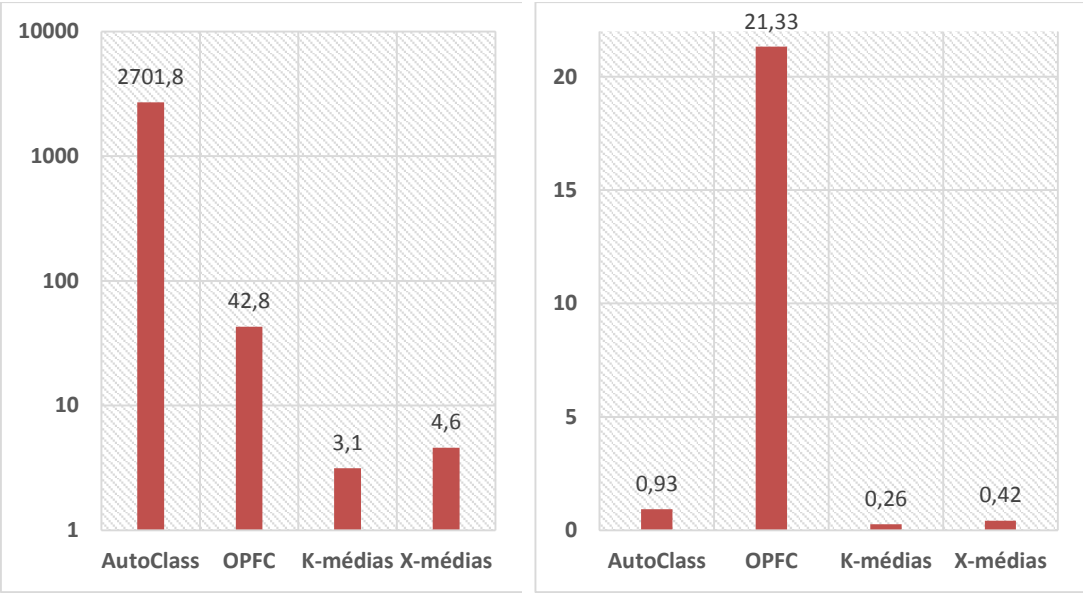
Outro importante aspecto a se considerar para a mensuração do desempenho dos algoritmos diz respeito aos tempos da etapa de treinamento e da etapa de classificação. Essa medida pode indicar a carga computacional exigida pelos algoritmos para o desempenho de suas funções. Para a classificação das anomalias, quanto menores os tempos de execução melhor.

Como ocorre para todo algoritmo de aprendizagem utilizado no contexto de tráfego de rede, a etapa de treinamento precisa ser realizada de tempos em tempos, de forma que o modelo de classificação possa acomodar as alterações no perfil do tráfego que normalmente ocorrem. Um longo tempo para a execução do treinamento pode representar um incômodo ao analista todas as vezes que este precisar ser realizado. Já com relação ao tempo da classificação, o impacto pode ser ainda maior. No caso de uma rede de grande porte, onde a taxa de emissão de alertas seja maior que a taxa de classificação em um dado intervalo, o sistema de classificação representaria um grande gargalo para a emissão dos alertas, o que poderia atrapalhar o processo de análise. O tempo[1] em segundos obtido por cada algoritmo é

[1] Neste trabalho, todos os tempos foram medidos com o método *currentTimeMillis* fornecido pela classe *Date* padrão da linguagem *Java.*

exibido na Figura 5-26. Na Figura 5-26 (a) é apresentado o tempo despendido com a etapa de treinamento e na Figura 5-26 (b) com a etapa de classificação. Com relação ao tempo despendido na etapa de treinamento, é possível ver que o tempo obtido pelo AutoClass foi bastante discrepante em relação ao obtido pelos outros algoritmos, em virtude disso escolheu-se a escala logarítmica para uma melhor representação.

Apesar disso, o AutoClass alcançou o melhor modelo de classificação ao se considerar a quantidade de classes, sendo somente 34 para a abordagem CFA. O menor tempo, como esperado, foi obtido pelo K-médias, já que que se trata da metodologia mais simples utilizada neste trabalho. O X-médias obteve um tempo levemente superior ao K-médias e o OPFC levou um tempo de 42,8 segundos. No entanto, o modelo obtido por este último foi o pior em termos do número de classes, com um total de 678 para a abordagem CFA.



(a) Tempo em segundos da etapa de treinamento – escala logarítmica.

(b) Tempo em segundos da etapa de classificação.

Figura 5-26. Tempos de execução de cada algoritmo.

No tocante ao tempo de classificação, é possível observar a vantagem daqueles algoritmos que alcançaram um bom modelo durante a fase de

treinamento. Como exemplo destaca-se o AutoClass, ainda que este tenha se mostrado o pior em termos do tempo de treinamento, devido ao seu bom modelo de classificação, obteve um tempo bastante consonante com os tempos obtidos pelos melhores algoritmos: K-médias e X-médias na fase de classificação. O OPFC, que gerou o pior modelo em termos do número de classes, obteve um tempo de classificação bastante superior à média dos outros algoritmos.

Dentre os algoritmos examinados, destaca-se novamente o X-médias que obteve tempos bastante similares àqueles obtidos pelo K-médias e demonstrou resultados médios de desempenho superiores a todos os algoritmos.

5.8 Redução de Falsos positivos

Outro resultado importante da classificação de anomalias é a possibilidade de identificar os falsos positivos erroneamente detectados pelo AD. Uma vez que se pode classificar os falsos positivos entre as classes mapeados para o tráfego normal, é possível economizar o tempo que analistas de segurança passariam analisando esses falsos alertas. No modelo de classificação proposto, as classes que majoritariamente agruparam as instâncias de tráfego legítimo durante o treinamento recebem o rótulo de Falsas Anomalias (FA).

Na etapa de classificação, uma anomalia pode ser considerada falsa quando é classificada entre as classes mapeadas como FA. Estas supostas falsas anomalias podem então ser eliminadas do conjunto de alertas, o que acarreta a redução do número de FPs considerados para os processos de análise.

Esta redução pode ser valiosa, pois permite uma melhora considerável na taxa de *Precisão*, inicialmente obtida pelo AD. No entanto, a desvantagem desta operação ocorre quando uma verdadeira anomalia, corretamente detectada, é erroneamente classificada como uma falsa anomalia e, portanto, eliminada da análise. Em tal situação, se notaria uma redução da taxa de *TVP*

inicialmente obtida pelo AD. O que significa que alguns ataques poderiam passar despercebidos, um grande problema para a detecção de intrusão. Apesar dessa possibilidade, pode-se ver que todos os algoritmos avaliados apresentaram resultados satisfatórios para as métricas de *Precisão* e *TVP* para as classes mapeadas como FA. Uma alta *TVP* para essas classes indica que a maioria das falsas anomalias foram corretamente classificadas, enquanto que uma boa *Precisão* indica que poucas ou nenhuma verdadeira anomalia foi erroneamente classificada como falsa.

Com a intenção de se avaliar o real impacto deste processo, foi simulada a eliminação de todas as anomalias classificadas nas classes FA, sejam elas verdadeiras anomalias ou de fato falsas. A eliminação de uma falsa anomalia causa a transformação de um caso FP num caso de VN, já a eliminação de uma verdadeira anomalia faz com que um caso de VP se transforme num caso de FN. Estas alterações causam distúrbios nas métricas de *TVP* e *Precisão*, observadas inicialmente. O objetivo desta redução é causar um aumento na *Precisão* ao custo da menor redução possível na *TVP*.

É clássico na literatura a relação inversa entre a *TVP* e *Precisão*, por isso é preciso se decidir qual métrica é mais importante, de acordo com o contexto da aplicação. Com o objetivo de se explorar esse desbalanço elencou-se as abordagens CA e CFA para o treinamento. Como visto nos resultados, com a abordagem CA os algoritmos apresentam resultados moderados na classificação dos falsos positivos com a vantagem de pouca ou nenhuma redução na *TVP* da detecção das anomalias. Já com a abordagem CFA, os algoritmos demonstram um desempenho bastante superior na classificação dos falsos positivos, entretanto, ao custo de uma redução um pouco maior na redução da *TVP*. A fim de se explorar este comportamento, é conveniente o uso da curva *Receiver Operating Characteristic* (ROC) (MAXION; ROBERTS, 2004) a qual relaciona o número de VP em função do número de FP. Na Figura 5-27 é apresentada uma curva ROC com os efeitos da redução dos alertas classificados como falsas anomalias, considerando-se os resultados da abordagem CA e CFA.

Na curva ROC, o eixo *y* é indicado pela *TVP*, já o eixo *x* é indicado pela *TFP*. É verificado que ao se considerar somente os resultados do AD, sem a

classificação com qualquer das abordagens, obtêm-se a *TVP* máxima, pois nenhum VP poderia ter sido adulterado. Em contrapartida, vê-se que a *TFP* é também máxima, visto que nenhum falso positivo foi eliminado. Ainda na curva, nos pontos correspondentes a abordagem CFA é verificada a menor *TFP* possível e um pouco de redução na *TVP*. Já nos respectivos pontos da abordagem CA, observa-se um pequeno acréscimo na *TVP*, entretanto é observado um considerável aumento na *TFP*. Na Figura 5-28 é apresentada uma versão ampliada da Figura 5-27 nos pontos de interesse das abordagens em análise.

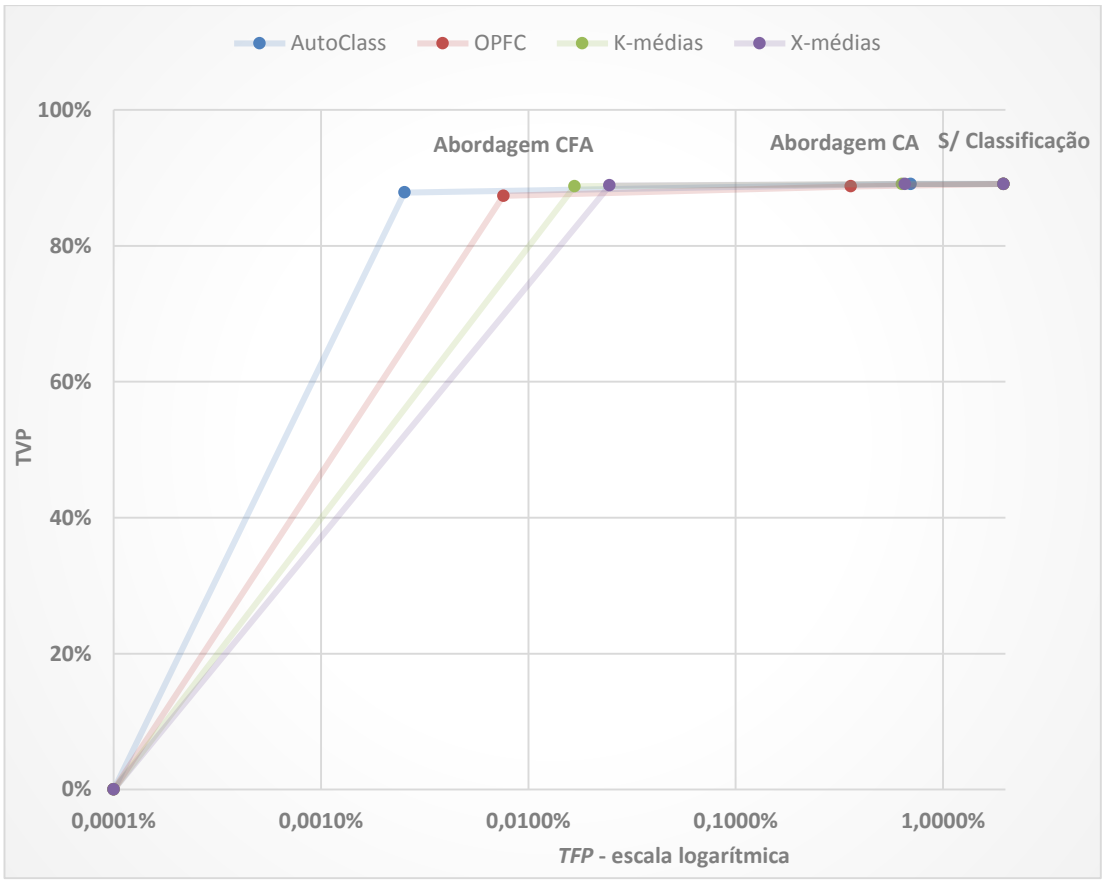


Figura 5-27. Curva ROC com resultados da redução de FPs.

Posto que é de interesse maiores valores para *TVP* e menores para *TFP*, é interessante que os valores de *TVP* cresçam num ritmo mais rápido que os valores de *TFP*. Assim, quanto mais uma curva ROC é acentuada em direção ao canto superior esquerdo, maior é o desempenho geral do modelo representado por ela. Tal característica é observada na Figura 5-27, o que

valida a qualidade do modelo de redução de falsos positivos. Além disso, o baixo crescimento da curva a partir dos pontos da abordagem CFA novamente comprova a qualidade do modelo obtido com esta abordagem, uma vez que o pequeno aumento da *TVP* pela abordagem CA não se torna atrativo diante do grande aumento da *TFP*.

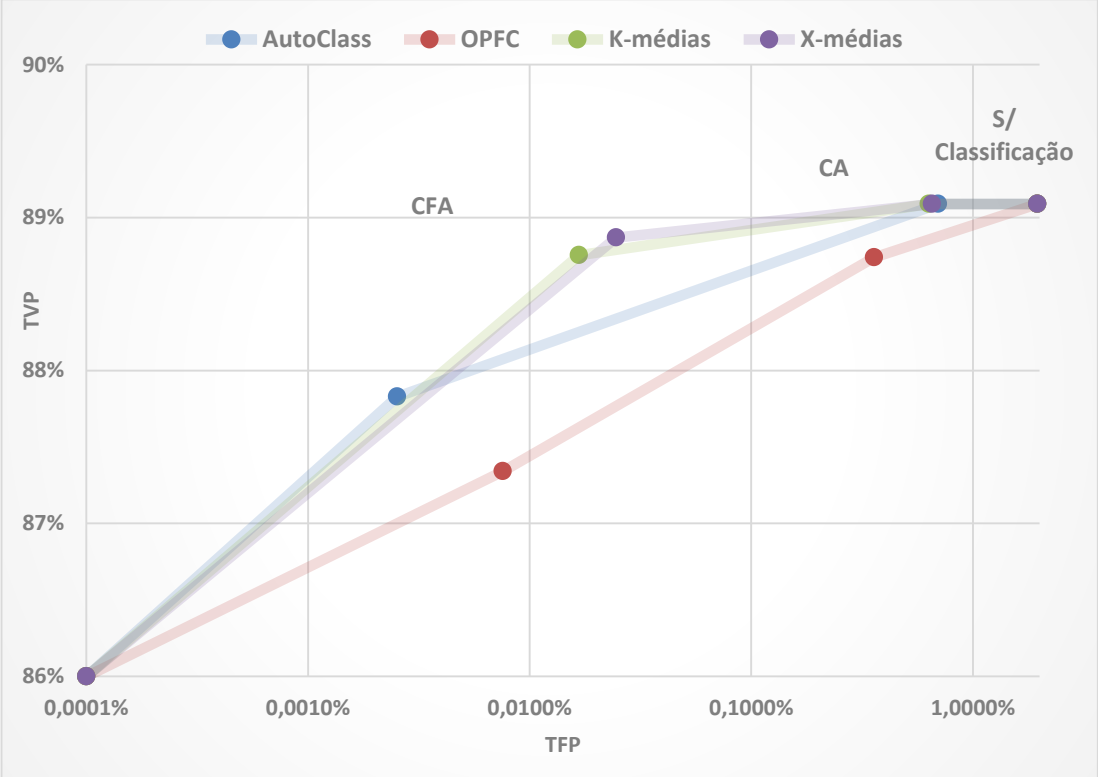


Figura 5-28. Figura 5-27 ampliada em pontos de interesse.

Ao se tratar do desempenho de cada algoritmo especificamente, é possível observar o clássico balanço que existe entre as taxas de verdadeiros positivos e falsos positivos. Como visto na curva o ROC, o AutoClass foi o algoritmo mais capaz na redução dos falsos positivos, entretanto este foi também foi o algoritmo que mais cometeu erros na classificação dos verdadeiros alertas, transformando detecções corretas em falsas anomalias. Em segurança da informação, os analistas sempre se deparam com esta situação, sendo necessário que se faça uma escolha entre priorizar a detecção de verdadeiros alertas e tolerar um maior número de falsos positivos; ou uma detecção com menos falsos positivos ao preço de se perder alguns verdadeiros alertas. De forma geral, costuma-se optar pela máxima

detecção dos verdadeiros alertas ao prejuízo de se aceitar alguns falsos positivos, visto ser extremante prejudicial a não identificação de verdadeiros ataques. Atendendo a este critério, o X-médias se mostrou o algoritmo mais adequado, ao ser aquele que menos adulterou detecções corretas realizadas pelo NIDS.

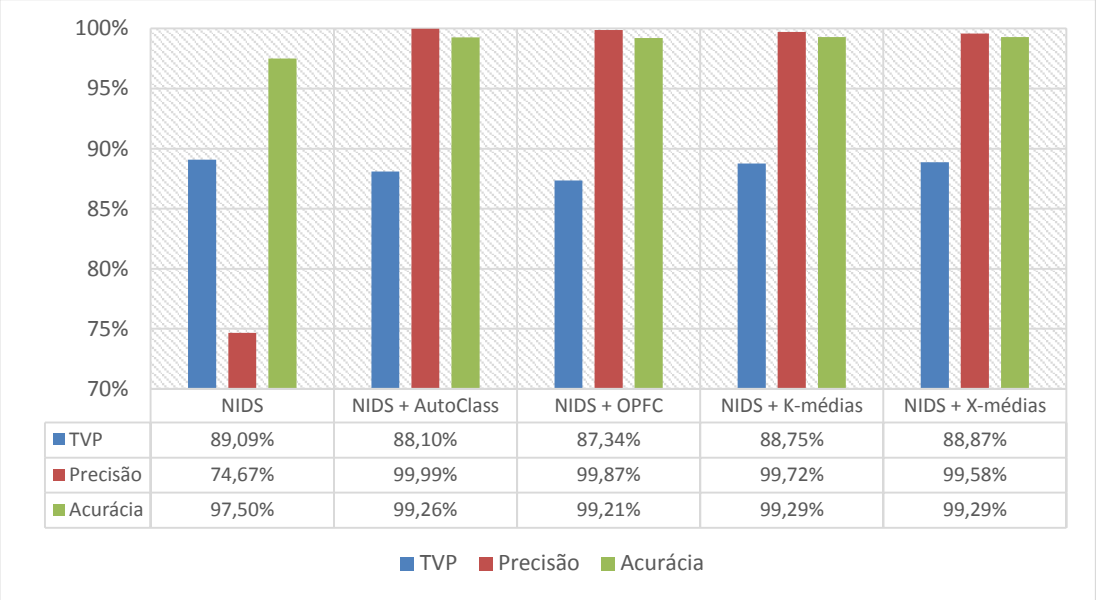


Figura 5-29. Métricas verificadas após a redução dos falsos positivos.

Considerando a eficácia demonstrada com a abordagem CFA, recalculou-se as métricas inicias da detecção, exibidas na Tabela 5-2, de acordo com os novos resultados obtidos com a classificação de cada algoritmo. Esses dados são apresentados na Figura 5-29. Nela pode ser verificado o benefício geral da classificação com a utilização de todos os algoritmos, visto que a *Precisão* foi aumentada de 74,67% para mais de 99% em todos os casos. Ao se considerar o AutoClass, por exemplo, conseguiu-se uma *Precisão* de 99,99%, indicando que dos 6.330 FPs, 6.328 foram corretamente eliminados. O custo desta redução foi a eliminação errônea de 207 verdadeiros positivos, o que causou uma queda na *TVP* de 89,09% para 88,10%. Ao se considerar o X-médias, pode ser observada a menor redução da *TVP*, pois somente 46 VPs foram eliminados ao benefício da eliminação de 6252 FPs. De fato, o algoritmo X-médias, juntamente com o K-médias, alavancou a Acurácia para o valor máximo de 99,29%. Com isso, puderam

ser confirmados os benefícios da redução dos FPs a partir da classificação de anomalias.

5.9 Considerações finais

Neste capítulo foram apresentados os resultados da classificação de anomalias e a consequente redução de falsos positivos que se pode realizar. Foram apresentados quatro diferentes métodos de agrupamento: AutoClass, OPFC, K-médias e X-médias os quais foram avaliados em duas abordagens diferentes de treinamento, a abordagem CA e a CFA.

Em termos gerais, todos os métodos apresentaram bons resultados, mas pode-se destacar o algoritmo X-médias que obteve melhores resultados na maioria dos experimentos. Ademais, todos os algoritmos demonstraram resultados harmoniosos de acordo com as diferentes abordagens utilizadas. Por mais que se tenha notado uma certa complementaridade entre as abordagens CA e CFA, a abordagem CFA se mostrou mais eficaz e por isso foi utilizada na apresentação dos resultados de comparação. No próximo capítulo são feitas as conclusões deste trabalho.

CAPÍTULO 6 - Conclusões

6.1 Conclusões gerais

Os NIDSs são ferramentas essenciais para se assegurar um maior nível de segurança às redes de computadores. No entanto, esta tecnologia, especialmente aquelas baseadas em anomalia, possuem algumas desvantagens como a dificuldade na identificação das anomalias detectadas e o elevado número de falsos positivos detectados. Neste sentido, esta pesquisa teve por objetivo estudar a classificação de anomalias como resposta às principais dificuldades dos NIDSs enquadrados nesta categoria.

Para tal, arquitetou-se o AnoID, um *framework* baseado em métodos de agrupamento para a classificação das anomalias detectadas por um NIDS na forma de alertas. Para a escolha do melhor método a ser utilizado, considerou-se os algoritmos AutoClass, OPFC, K-médias e X-médias, os quais foram avaliados de acordo com sua capacidade em classificar as anomalias, bem como os falsos positivos erroneamente detectados. Além dos algoritmos elencados, foram estabelecidas as abordagens CA e CFA para o treinamento dos métodos. O objetivo inicial foi a obtenção de dois modelos, um com maior eficácia na classificação das verdadeiras anomalias (CA) e outro com maior desempenho na classificação dos falsos positivos (i.e., falsas anomalias) (CFA). Entretanto a abordagem CFA mostrou resultados bem competitivos com a abordagem CA na classificação dos alertas verdadeiros, ao benefício

de um desempenho bastante superior na identificação e redução dos falsos positivos.

Com relação aos métodos avaliados, observou-se a eficácia de todos os algoritmos na classificação de anomalias, visto que todos obtiveram uma *Acurácia Global* de mais de 97%. Contudo, o X-médias se destacou em diversos aspectos como na obtenção da máxima *Acurácia Global*, a máxima Medida-F e por ser o algoritmo que menos adulterou detecções corretas pelo NIDS escolhido. Além disso, o X-médias alcançou um bom tempo, tanto para a etapa de treinamento como para a classificação, ficando atrás somente do algoritmo K-médias, que é geralmente tido como o método de agrupamento mais simples e portando mais rápido.

Além da classificação propriamente dita, esta pesquisa também se propôs a mostrar os benefícios que tal classificação pode fornecer na identificação de falsos positivos erroneamente detectados pelos NIDSs. Neste trabalho simulou-se a remoção de todas as anomalias classificadas pelo sistema como falsas. Como resultado foi possível alavancar a taxa de *Precisão* da fase de detecção de 74,67% para mais de 99%. Esse resultado representa um importante avanço, uma vez que permite aos analistas despenderem seu tempo, na maior parte, com alertas importantes de verdadeiros ataques. Ademais, com um número menor de falsos positivos cria-se um terreno mais seguro para a aplicação de contramedidas de forma automática.

Em resumo, este trabalho contribui com uma comparação entre diferentes métodos de agrupamento para a classificação de anomalias, colaborando com a literatura que até então considerou os métodos supervisionados em sua maioria. Além disso, mostrou-se o potencial desta classificação também para a redução dos falsos positivos detectados pelos NIDSs. Portanto, conclui-se que este projeto obteve sucesso em relação àquilo que foi proposto, seus objetivos foram integralmente cumpridos e se contribuiu em caráter inovador com a comunidade de pesquisa em segurança.

6.2 Trabalhos futuros

Uma característica do modelo de classificação proposto é que este foi arquitetado para a classificação do comportamento anômalo gerado por um certo endereço de origem na rede. No entanto, alguns NIDSs focam na detecção de anomalias em determinados serviços e não centralizam suas detecções nas origens dos atacantes (PROTO et al, 2010). Essas abordagens são úteis na detecção de ataques distribuídos como por exemplo ataques distribuídos de negação de serviço. Uma proposta de trabalho futuro seria arquitetar uma classificação também para esse tipo de detecção. Para isso seria necessária uma adequação das características apresentadas na

Tabela 4-1, uma vez que, atualmente, estas devem ser agrupadas de acordo com cada endereço de origem no tráfego.

Uma outra proposta consiste na modelagem de classes especiais para a classificação de anomalias desconhecidas para o sistema. Assim, além de classes para os ataques e falsas anomalias, haveriam classes para as anomalias não consideradas durante a fase de treinamento. Uma hipótese para tal, seria a consideração de uma porção especial de tráfego com características discrepantes para a etapa de treinamento, isso causaria a geração de classes especiais que poderiam atrair a classificação desses eventos.

Finalmente, propõe-se a expansão do conjunto de ataques considerados, o que poderia conferir maior versatilidade em relação às anomalias classificáveis pelo AnoID.

6.3 Dificuldades encontradas

Durante o desenvolvimento desta pesquisa deparou-se com várias dificuldades que precisaram ser superadas. Dentre elas, destaca-se a definição das características relevantes para a classificação. A dificuldade maior da classificação proposta é promover uma diferenciação entre as taxonomias individuais de anomalias e não somente entre o tráfego legítimo

e o anômalo. Para esta definição, avaliou-se diferentes conjuntos de características de acordo com estudos sobre o comportamento dos eventos considerados, ao fim deste processo escolheu-se as características apresentadas na

Tabela 4-1 que propiciaram bons resultados nos testes com todos os métodos avaliados.

Outra dificuldade foi com relação à normalização das características para a obtenção dos melhores resultados possíveis com os algoritmos avaliados. Observou-se um grande impacto deste processo sobre os algoritmos, em determinados casos positivo, em outros casos, negativo. A fim de se encontrar um modelo de normalização que potencializasse os resultados dos algoritmos utilizados, foram realizados vários experimentos com base em modelos observados em outros trabalhos na literatura.

Por fim, algumas dificuldades também foram enfrentadas ao se utilizar diferentes algoritmos de agrupamento, já que cada algoritmo define seu padrão de entrada e saída de dados. Portanto, foi necessário o desenvolvimento de módulos com o objetivo de adequar os dados ao padrão aceito por cada algoritmo utilizado, além de processar os dados de saída nos diferentes formatos por eles definidos.

6.4 Produções

Este trabalhou produziu como resultado dois artigos. O primeiro, intitulado "*A model for anomaly classification in intrusion detection systems*", foi publicado no "*Journal of Physics: Conference Series (JPCS)*". O segundo, com título: "*Anomalies Classification and False Positives Reduction in Network Intrusion Detection Systems through Optimum-Path Forest Clustering*" foi submetido ao "*Transactions on Knowledge Discovery from Data*" e encontra-se em revisão no momento desta escrita.

Ademais, este trabalho foi apresentado no V Workshop do Programa de Pós-Graduação em Ciência da Computação da UNESP (V WPPGCC-

UNESP), onde recebeu o título de melhor trabalho apresentado na linha de pesquisa: “Arquitetura de Computadores e Sistemas Distribuídos”.

Referências Bibliográficas

AHMED, M.; MAHMOOD, A. N.; HU, J. A survey of network anomaly detection techniques. Journal of Network and Computer Applications, v. 60, p. 19–31, jan. 2016.

AHMED, M.; MAHMOOD, A. N. Network traffic analysis based on collective anomaly detection. In: 2014 9th IEEE Conference on Industrial Electronics and Applications, jun. 2014.

Arachni. Arachni - Web Application Security Scanner Framework. Disponível em: <http://www.arachni-scanner.com/>. Acesso em: 01 abr. 2016.

AutoClass C. What is AutoClass - National Aeronautics and Space Administration. Disponível em: <https://ti.arc.nasa.gov/tech/rse/synthesis-projects-applications/autoclass/autoclass-c/>. Acesso em: 01 abr. 2016.

BATISTA, M. L. Análise de eventos de segurança em redes de computadores utilizando detecção de novidade. 2012. 72 f. Dissertação (Mestrado em Ciência da Computação) – Instituto de Biociências, Letras e Ciências Exatas, Universidade Estadual Paulista, São José do Rio Preto, 2012.

BATISTA, M, L.; CANSIAN, A, M. Detecção de eventos em redes de computadores utilizando detecção de novidade. IADIS Ibero Americana WWW/Internet. Rio de Janeiro, RJ, Brazil. 2011.

CENTRO DE ESTUDOS, RESPOSTA E TRATAMENTO DE INCIDENTES DE SEGURANÇA NO BRASIL (CERT.Br). Estatísticas sobre notificações de incidentes. Disponível em: <http://www.cert.br/stats/>. Acesso em: 04 ago. 2015.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. ACM Computing Surveys, v. 41, n. 3, p. 1–58, 2009.

CHEESEMAN, P.; STUTZ, J. Bayesian classification (AutoClass): theory and results. Advances in knowledge discovery and data mining. Menlo Park. p.153–180, 1996.

CHUNG, C.; KHATKAR, P.; XING, T.; LEE, J.; HUANG, D. NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems. IEEE Transactions on Dependable and Secure Computing, v. 10, n. 4, p. 198–211, jul. 2013.

CLAISE, B. (Ed.). RFC 5101: specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. Disponível em: <http://tools.ietf.org/html/rfc5101/>. Acesso em: 01 abr. 2016.

COSTA, K. A. P.; PEREIRA, L. A. M.; NAKAMURA, R. Y. M.; PEREIRA, C. R.; PAPA, J. P.; XAVIER FALCÃO, A. A nature-inspired approach to speed up optimum-path forest clustering and its application to intrusion detection in computer networks. Information Sciences, v. 294, p. 95–108, fev. 2015.

DEMPSTER, A. P. A.; LAIRD, N. M. N.; RUBIN, D. D. B. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society Series B Methodological, v. 39, n. 1, p. 1–38, 1977.

DONG, W.; MOSES, C.; LI, K. Efficient k-nearest neighbor graph construction for generic similarity measures. In: Proceedings of the 20th international conference on World wide web - WWW '11, New York, New York, USA. Anais... New York, New York, USA: ACM Press, 2011.

ELSHOUSH, H. T.; OSMAN, I. M. Reducing false positives through fuzzy alert correlation in collaborative intelligent intrusion detection systems - A review. In: International Conference on Fuzzy Systems. p. 1-8, jul. 2010.

ERMAN, J.; MAHANTI, A.; ARLITT, M. Internet Traffic Identification using Machine Learning. Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE, p.1-6, 2006a.

ERMAN, J.; ARLITT, M.; MAHANTI, A. Traffic classification using clustering algorithms. In: Proceedings of the 2006 SIGCOMM workshop on Mining network data - MineNet '06. New York, New York, USA: ACM Press, 2006b.

FERNANDES, G.; OWEZARSKI, P. Automated classification of network traffic anomalies. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, v. 19 LNICST, p. 91–100, 2009.

Fprobe. NETWORK UPTIME - THE ONLINE RESOURCE FOR NETWORK PROFESSIONALS. Disponível em: <http://www.networkuptime.com/tools/netflow/fprobe.html>. Acesso em: 01 abr. 2016.

FONSECA, J.; VIEIRA, M.; MADEIRA, H. "Testing and Comparing Web Vulnerability Scanning Tools for SQL Injection and XSS Attacks," Dependable Computing. PRDC 2007. 13th Pacific Rim International Symposium on, p.365-372, 17-19. 2007.

GENTLEMAN, R.; CAREY, V. J. Unsupervised Machine Learning. Bioconductor Case Studies, Springer New York. p. 137-157, 2008.

Hydra. THC Hydra. Disponível em: <http://sectools.org/tool/hydra/>. Acesso em: 01 abr. 2016.

JEONG, H.; YOO, Y.; YI, K. M.; CHOI, J. Y. Traffic Pattern Analysis and Anomaly Detection via Probabilistic Inference Model. In: KYUNG, C.-M. (Ed.). Theory and Applications of Smart Cameras. Dordrecht: Springer Netherlands, 2016. p. 215–240.

KVM. Kernel Virtual Machine. Disponível em: < http://www.linux-kvm.org/page/Main_Page>. Acesso em: 01 abr. 2016.

LAKHINA, A.; CROVELLA, M.; DIOT, C. Mining anomalies using traffic feature distributions. Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '05, p. 217, 2005.

LibOPF. OPF Classifiers Library. Disponível em: < http://www.ic.unicamp.br/~afalcao/libopf/>. Acesso em: 01 abr. 2016.

LIN, W. C.; KE, S. W.; TSAI, C. F. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. Knowledge-Based Systems, v. 78, p. 13–21, abr. 2015.

MARTINS, G. B.; AFONSO, L. C. S.; OSAKU, D.; ALMEIDA, J.; PAPA, J. P. Static Video Summarization through Optimum-Path Forest Clustering. In: BAYRO-CORROCHANO, E.; HANCOCK, E. (Ed.). Cham: Springer International Publishing. p. 893–900. 2014.

MAXION, R. A.; ROBERTS, R. R. Proper Use of ROC Curves in Intrusion / Anomaly Detection.School of Computing Science, University of Newcastle upon Tyne. 2004.

MIRKIN, B. (2005). Clustering for Data Mining: A Data Recovery Approach. Chapman & Hal. Citado nas páginas 34 e 93.

MIRSHAHJAFARI, M.; GHAVAMNIA, H. Classifying IDS alerts automatically for use in correlation systems. 2014 11th International ISC Conference on Information Security and Cryptology, p. 126–130, set. 2014.

MUDA, Z.; YASSIN, W.; SULAIMAN, M. N.; UDZIR, N. I. Intrusion detection based on K-Means clustering and Naive Bayes classification. In: 2011 7th International Conference on Information Technology in Asia, Anais...IEEE, jul. 2011.

NGUYEN, T.; ARMITAGE, G. A survey of techniques for internet traffic classification using machine learning. IEEE Communications Surveys & Tutorials, v. 10, n. 4, p. 56–76, 2008.

Nmap. Nmap Security Scanner. Disponível em: < https://nmap.org/>. Acesso em: 01 abr. 2016.

PAPA, J. P.; FALCÃO, A. X.; SUZUKI, C. T. N. Supervised pattern classification based on optimum-path forest. International Journal of Imaging Systems and Technology, v. 19, n. 2, p. 120–131, jun. 2009.

PAREDES-OLIVA, I.; CASTELL-UROZ, I.; BARLET-ROS, P.; DIMITROPOULOS, X.; SOLÉ-PARETA, J. Practical anomaly detection based on classifying frequent traffic patterns. In: Proceedings - IEEE INFOCOM, Anais...2012.

PAXSON, V. Empirically derived analytic models of wide-area TCP connections. IEEE/ACM Transactions on Networking (TON), v. 2, n. 4, p. 316–336, 1994.

PEREIRA, C. R.; NAKAMURA, R. Y. M.; COSTA, K. a. P.; PAPA, J. P. An Optimum-Path Forest framework for intrusion detection in computer networks. Engineering Applications of Artificial Intelligence, v. 25, n. 6, p. 1226–1234, set. 2012.

PROTO, A. Detecção de eventos de segurança de redes por intermédio de técnicas estatísticas e associativas aplicadas a fluxos de dados. 2011. 73 f. Dissertação (Mestrado em Ciências da Computação–Instituto de Biociências, Letras e Ciências Exatas, Universidade Estadual Paulista, São José do Rio Preto, 2011.

PROTO, A.; ALEXANDRE, L. A.; Batista M. L.; OLIVEIRA, I. L.; CANSIAN, A. M. Statistical Model Applied to *NetFlow* for Network Intrusion Detection. Transactions on Computational Science, v. 6480, p. 179-191, 2010.

RINGBERG, H.; ROUGHAN, M.; REXFORD, J. The need for simulation in evaluating anomaly detectors. ACM SIGCOMM Computer Communication Review, v. 38, n. 1, p. 55, 2008.

ROCHA, L. M.; CAPPABIANCO, F. A. M.; FALCÃO, A. X. Data clustering as an optimum-path forest problem with applications in image analysis. International Journal of Imaging Systems and Technology, v. 19, n. 2, p. 50–68, jun. 2009.

SHI, J.; MALIK, J. Normalized Cuts and Image Segmentation. Ieee Transactions on Pattern Analysis and Machine Intelligence, v. 22, n. 8, p. 888–905, 2000.

TELLENBACH, B.; BURKHART, M.; SCHATZMANN, D.; GUGELMANN, D.; SORNETTE, D. Accurate network anomaly classification with generalized entropy metrics. Computer Networks, v. 55, n. 15, p. 3485–3502, out. 2011.

TREURNIET, J. Detecting low-profile scans in TCP anomaly event data. In: Proceedings of the 2006 International Conference on Privacy, Security and Trust Bridge the Gap Between PST Technologies and Business Services - PST ’06, New York, New York, USA. Anais... New York, New York, USA: ACM Press, 2006.

WITTEN, I. H.; FRANK, E. 2000. Data mining: practical machine learning tools and techniques with java implementations. New York, NY, USA: Morgan Kaufmann Publishers. 629.

WU, S. X.; BANZHAF, W. The use of computational intelligence in intrusion detection systems: A review. Applied Soft Computing, v. 10, n. 1, p. 1–35, Jan. 2010.

XING, T.; HUANG, D.; XU, L.; CHUNG, C.-J.; KHATKAR, P. SnortFlow: A OpenFlow-Based Intrusion Prevention System in Cloud Environment. In: 2013 Second GENI Research and Educational Experiment Workshop, Anais...IEEE, mar. 2013.

ZANDER, S; NGUYEN, T.T.T.; ARMITAGE, G. Automated traffic classification and application identification using machine learning, Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on, p.250-257, 17-17 Nov. 2005.

ZHANG, J.; LI, H.; GAO, Q.; WANG, H.; LUO, Y. Detecting anomalies from big network traffic data using an adaptive detection approach. Information Sciences, v. 318, p. 91–110, out. 2015.