Vinícius Vassoler Galhardi

Detecção adaptativa de anomalias em redes de computadores utilizando técnicas não supervisionadas

São José do Rio Preto
2017

Vinícius Vassoler Galhardi

Detecção adaptativa de anomalias em redes de computadores utilizando técnicas não supervisionadas

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
2017

Galhardi, Vinícius Vassoler.
Detecção adaptativa de anomalias em redes de computadores utilizando técnicas não supervisionadas / Vinícius Vassoler Galhardi. -- São José do Rio Preto, 2017
71 f. : il., gráfs., tabs.

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

1. Ciência da Computação. 2. Redes de computadores – Medidas de segurança. 3. Sistemas de detecção de intrusão (Medidas de segurança) 4. Detecção de anomalias (Medidas de segurança) 5. Aprendizado do computador. 6. Fluxo de dados (Computadores) I. Universidade Estadual Paulista "Júlio de Mesquita Filho". Instituto de Biociências, Letras e Ciências Exatas. II. Título.

CDU – 681.3.025

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

Vinícius Vassoler Galhardi

Detecção adaptativa de anomalias em redes de computadores utilizando técnicas não supervisionadas

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

Comissão Examinadora

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

Prof. Dr. Cesar Augusto Cavalheiro Marcondes
Universidade Federal de São Carlos

Prof. Dr. Leandro Alves Neves
UNESP – São José do Rio Preto

São José do Rio Preto
18 de janeiro de 2017

AGRADECIMENTOS

Agradeço primeiramente a Deus por Sua graça e todas as oportunidades em minha vida.

À minha esposa Michele por todo amor, carinho e compreensão, estando ao meu lado em todos os momentos, além de tudo o que já construímos juntos. Agradeço também minha família pelo amor, ensinamento e apoio durante todos estes anos.

Ao meu orientador, Prof. Dr. Adriano Mauro Cansian, por sua orientação, contribuindo significativamente para minha formação acadêmica, e todos os conselhos profissionais e pessoais.

Aos meus amigos Raphael Campos Silva e Vinícius Oliveira Ferreira, que estiveram comigo nessa jornada desde o início, por todos os momentos juntos durante esta etapa de nossas vidas.

Aos colegas de laboratório: Adriano Ribeiro, Amanda Barbosa, Bruno Leal, Leandro Gonçalves, Matheus Andrade, Pedro Ferracini e Rafael Carreira pelas horas de estudo e momentos divertidos que passamos. Agradeço também aos membros que estiveram lá antes de mim e passaram grandes conhecimentos.

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

Por fim, agradeço à CAPES pela bolsa de mestrado concedida para realização deste projeto.

“Na vida, não vale tanto o que temos, nem tanto importa o que somos. Vale o que realizamos com aquilo que possuímos e, acima de tudo, importa o que fazemos de nós!”

Chico Xavier

RESUMO

Ataques às redes de computadores têm sido cada vez mais constantes e possuem grande capacidade destrutiva. Os sistemas de detecção de intrusão possuem um importante papel na detecção destas ameaças. Dentre estes sistemas, a detecção de anomalias tem sido uma área amplamente explorada devido à possibilidade de detectar ataques até então desconhecidos. Devido à complexidade para a geração de modelos que sejam capazes de descrever o comportamento padrão de um ambiente, técnicas de aprendizagem automática vêm sendo amplamente exploradas. Este trabalho aborda a detecção de ataques a redes de computadores utilizando uma combinação de técnicas de agrupamento. Desse modo, espera-se obter um sistema adaptativo, capaz de encontrar anomalias presentes na rede sem a necessidade de uma etapa de treinamento com dados rotulados. Dado que a taxa de falsos negativos é um dos maiores problemas encontrados na utilização de algoritmos não supervisionados, pretende-se alcançar uma melhora neste quesito através do uso combinado de diferentes técnicas.

Palavras-chave: Detecção de anomalias em redes. Aprendizado não supervisionado. Análise de fluxo de dados.

ABSTRACT

Attacks on computer networks have been constantly increased and have great destructive capacity. Intrusion detection systems have an important role in the detection of these threats. Among these systems, anomaly detection has been widely explored due to the possibility of detecting unknown attacks. These systems are usually built using machine learning techniques due to the complexity of generating models capable of describing the normal behavior of an environment. We aim to addresses the detection of anomalies on computer networks using a combination of clustering techniques. Thus, we expect to achieve an adaptive system, able to find anomalies present in the network without the need of a training step with labeled data. Given that false positive rate is one of the major problems faced when using unsupervised algorithms, we intend to achieve an improvement in this issue with the combined use of different techniques.

Keywords: Network anomaly detection. Unsupervised learning. Data stream analysis.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACC: *Ant Colony Clustering*
ACO: *Ant Colony Optimization*
AP: *Affinity Propagation*
APT: *Advanced Persistent Threat*
CAIDA: *Center of Applied Internet Data Analysis*
DBSCAN: *Density-based Spatial Clustering of Applications with Noise*
DPI: *Deep Packet Inspection*
HTTP: *Hypertext Transfer Protocol*
IDS: *Intrusion Detection System*
IETF: *Internet Engineering Task Force*
IP: *Internet Protocol*
IPFIX: *IP Flow Information Export*
PCA: *Principal Component Analysis*
PSO: *Particle Swarm Optimization*
RDD: *Resilient Distributed Dataset*
ROC: *Receiver Operating Characteristic*
SKL: *Sequential Karhunen–Loeve*
SOM: *Self-organizing Map*
SVM: *Support Vector Machine*

SUMÁRIO

CAPÍTULO 1 - Introdução

1.1 Considerações iniciais

Atualmente há uma utilização massiva de serviços disponíveis através da Internet, como *e-commerce*, redes sociais, *internet banking*, dentre outros. Como consequência desta comodidade e praticidade, o volume de informações sensíveis trafegando através das redes de computadores e armazenadas nos sistemas computacionais não para de crescer. Aliado à disponibilidade de informações sensíveis, há o lançamento de aplicações sem qualquer preocupação em incorporar práticas de segurança durante seu desenvolvimento, tornando estas aplicações vulneráveis a ataques. Na tentativa de tirar proveito deste cenário, ataques visando comprometer as informações e/ou sistemas são frequentes.

Dentre os principais incidentes reportados, muitos são conhecidos há tempos, como prospecção de redes (chamados genericamente de *Scans*), propagação de *worms*, tentativas de fraudes (*scams*), dentre outros. Tais ataques podem causar grandes prejuízos às organizações e, portanto, é necessária a utilização de mecanismos de detecção e prevenção de ataques a fim de evitar que esses tenham sucesso (CERT.BR, 2015).

Com o intuito de detectar e eventualmente coibir os ataques, diferentes mecanismos de proteção foram criados e são utilizados. Dentre esses mecanismos, os Sistemas de Detecção de Intrusão (IDS – *Intrusion Detection System*) são utilizados para o monitoramento e detecção de eventos

maliciosos. Os IDSs podem realizar a detecção de ameaças a partir de assinaturas de ataques bem conhecidos, nomeada detecção por abuso, ou a partir de anomalias presentes no ambiente monitorado. Apesar de apresentarem um baixo nível de falso-positivo, as metodologias baseadas em assinaturas só detectam ataques previamente conhecidos. Já as metodologias baseadas em anomalia têm a capacidade de detectar novos ataques, pois estas fazem sua detecção a partir de comportamentos que diferem do comportamento normal. No entanto, tais metodologias apresentam uma alta taxa de falso-positivo (WU; BANZHAF, 2010).

Ainda, a detecção por abuso depende da criação de assinaturas a partir da análise de eventos maliciosos, ficando dependente do nível de conhecimento e esforço empenhado por parte dos analistas de segurança. Embora haja pesquisas no sentido de gerar as assinaturas de forma automática, tais sistemas ainda dependem da indicação se um determinado evento é malicioso ou não. Portanto, a detecção de anomalias é uma etapa inicial importante no processo de detecção de novas ameaças. Uma análise comparativa entre os dois métodos é apresentada por Elshoush e Osman (2011).

1.2 Identificação do problema e justificativa

Face ao constante crescimento do tráfego nas redes de computadores, o desenvolvimento de IDSs eficientes tem sido um desafio cada vez maior. É de extrema importância que a detecção de ataques seja feita nos seus estágios iniciais para que, possivelmente, seja aplicada alguma contramedida.

Dentre as duas classes nas quais os IDSs são classificados por seu método de detecção, a área de detecção por anomalia tem sido a mais explorada pelos pesquisadores recentemente (CHEN et al., 2014). Devido à complexidade para a geração de modelos que sejam capazes de descrever o comportamento padrão de um ambiente, seja uma rede de computadores ou ações realizadas em um único *host*, técnicas de aprendizagem automática vêm sendo amplamente exploradas. A utilização de algoritmos de

aprendizado de máquina implica em um processo de treinamento para a geração do modelo. O processo de aprendizado, ou criação do modelo de agrupamento, necessita de um conjunto de dados de treinamento, onde cada item deste conjunto pode ser representado por um vetor de características.

Ao propor novos métodos e ferramentas, é necessário considerar o crescimento exponencial da quantidade e dimensionalidade dos dados (FENG et al., 2014). Outro problema encontrado é que o sistema deve ser capaz de se adaptar a possíveis mudanças que possam ocorrer no ambiente monitorado, tornando o modelo defasado, exigindo um novo treinamento do algoritmo.

Dessa forma, as proposições de novas metodologias para detecção de intrusão devem considerar os desafios intrínsecos ao tratamento de grandes volumes de dados. Dependendo do tamanho do conjunto de treinamento, o processo de extração de características pode se tornar muito demorado. Para tal, faz-se necessário o uso de técnicas que tornem esse processo escalável, possibilitando assim, aumentar a capacidade de processamento de grandes conjuntos de dados.

1.3 Hipótese e objetivos

Este trabalho aborda a detecção de ataques a redes de computadores utilizando uma combinação de técnicas de agrupamento. Desse modo, é proposto um sistema adaptativo, capaz de encontrar anomalias presentes na rede sem a necessidade de uma etapa de treinamento com dados rotulados. Uma vez que a taxa de falsos negativos é um dos maiores problemas encontrados na utilização de algoritmos não supervisionados, pretende-se alcançar uma melhora neste quesito através do uso de técnicas combinadas.

Ainda, com o objetivo de tornar o sistema mais escalável e capaz de lidar com grande volume de dados, o método proposto realiza suas adaptações de modo incremental, visando lidar com os desafios da análise de fluxos contínuos de dados.

1.4 Organização da dissertação

Esta dissertação está dividida em seis capítulos, considerando este apresentado. No capítulo 2 é realizada uma fundamentação teórica, a qual se refere aos conceitos necessários para a compreensão do trabalho e as tecnologias utilizadas. No capítulo 3 são apresentados os trabalhos relacionados utilizados como base para este trabalho. No capítulo 4 é apresentada uma descrição da metodologia utilizada. No capítulo 5 os experimentos e seus resultados são discutidos. Finalmente, no capítulo 6 são apresentadas as considerações finais.

CAPÍTULO 2 - Fundamentação teórica

2.1 Considerações iniciais

Neste capítulo é apresentada a fundamentação teórica referente aos conceitos e tecnologias utilizados nesse trabalho. Na seção 2.2 são apresentados alguns ataques comuns a redes de computadores, se aprofundando naqueles utilizados nos testes realizados. Na seção 2.3 são apresentados os conceitos por traz de um sistema de detecção de intrusão e as suas principais classificações. Na seção 2.4 é apresentado o conceito de fluxo de dados em redes de computadores, explicitando os padrões *NetFlow* e *Biflow*. Por fim, na seção 2.5 são apresentadas algumas metodologias de aprendizado de máquina utilizados na literatura em sistemas de detecção de intrusão por anomalia.

2.2 Segurança de Computadores e Redes

A área de segurança da informação contempla a pesquisa de metodologias e estratégias com objetivo de proteger e preservar informações sensíveis e sistemas computacionais.

A manutenção e garantia desses requisitos é fundamental para a segurança de sistemas computacionais que, uma vez comprometidos, podem apresentar grandes prejuízos. Portanto, é de absoluta importância a garantia

de que os sistemas computacionais sejam seguros contra as constantes ameaças que existem.

Ataques a redes de computadores têm se tornado um problema para a sociedade, podendo causar diversos prejuízos, principalmente financeiros. Existem diversos tipos de ataques e estes estão em constante evolução, porém, há alguns que são considerados básicos e possuem grande importância na fundamentação de ataques mais elaborados. Nesta seção são apresentados brevemente alguns ataques de interesse deste projeto e alguns eventos que possivelmente podem preceder um ataque.

2.2.1 Mapeamento de rede

O mapeamento de rede pode ser considerado como a fase inicial de um ataque, em que é feito o reconhecimento do ambiente alvo. Tem como objetivo a descoberta de *hosts* ativos em uma rede e é também conhecido como varredura de rede, *scan* ou prospecção de rede. O resultado de um *scan* de rede permite a um atacante definir os alvos ativos e prosseguir com os passos subsequentes, como, por exemplo, descobrir quais serviços um *host* ativo está executando. A detecção de um mapeamento de rede pode auxiliar um analista a aplicar métodos de contramedidas para eventuais ataques que possam suceder o mapeamento.

2.2.2 Mapeamento de serviços

O mapeamento ou *scan* de serviços é utilizado para descobrir quais serviços estão em execução em um determinado *host*. É comum a sua utilização após o mapeamento de rede e normalmente é utilizado para a descoberta de serviços possivelmente vulneráveis.

Entretanto, os mapeamentos de redes e serviços são comumente utilizados para auditoria e administração de redes de computadores. Portanto,

ao perceber a ocorrência desses determinados eventos, deve-se verificar qual a origem e, posteriormente, aplicar alguma contramedida se necessário.

2.2.3 Ataques de dicionário e força bruta

Ataques de força bruta têm como objetivo a descoberta de senhas e pode ser caracterizado pela realização de diversas conexões rápidas e sequenciais em um mesmo serviço. As tentativas de acesso utilizam todas as possibilidades de formação de palavras, cobrindo um número muito grande de possibilidades. Os serviços mais visados são SSH, Telnet, FTP, entre outros. Uma variante mais otimizada deste tipo de ataque é o ataque de dicionário e seu funcionamento é utilizar uma lista de palavras, comumente utilizadas pelos usuários na composição de senhas, e fazer as tentativas de acesso utilizando estas palavras.

2.2.4 Negação de serviço

Ataques de negação de serviço, ou *Denial of Service* (DoS), são técnicas utilizadas para paralisar um serviço em execução ou tirar de operação um computador conectado à Internet utilizando apenas um computador (CERT.BR, 2015). Este tipo de ataque não tem como objetivo a coleta de informações ou a invasão de sistemas, seu objetivo é tornar o serviço indisponível através do esgotamento de recursos do alvo. Este tipo de ataque causa um prejuízo a todos os usuários que utilizam o recurso afetado.

Uma variação de DoS amplamente utilizada é o chamado ataque de negação de serviço distribuído, ou DDoS (*Distributed Denial of Service*). Um DDoS é realizado a partir de dois ou mais computadores, aumentando a capacidade computacional dos atacantes, além de dificultar a aplicação de contramedidas por parte das vítimas. É comum a utilização de *botnets* para a realização de ataques DDoS.

2.2.5 Códigos maliciosos

Códigos maliciosos, ou *malwares*, são programas desenvolvidos com o objetivo de executar atividades maliciosas em um sistema. A utilização de *malwares* é fortemente motivada à obtenção de vantagens financeiras, a coleta de informações confidenciais e o vandalismo (CERT.BR, 2015). Os principais tipos de *malwares* existentes são abordados a seguir.

- **Vírus:** é um programa ou parte de um programa malicioso que depende da execução do programa ou arquivo hospedeiro. Um vírus se propaga inserindo cópias de si mesmo em arquivos ou se tornando parte de outros programas.
- ***Worm*:** diferentemente dos vírus, os *worms* possuem a capacidade de se propagarem automaticamente pelas redes, enviando cópias de si mesmo para outros computadores. O processo de propagação de um *worm* envolve a identificação dos computadores alvos, por exemplo através de varreduras de rede; o envio das cópias, sendo uma das formas a exploração de vulnerabilidades existentes na máquina alvo; a ativação das cópias; e o reinício do processo, onde o *malware* passa a utilizar o computador que era vítima como fonte de propagação.
- ***Bots* e *botnet*:** um *bot* é um programa que possibilita o controle remoto da máquina infectada por parte do atacante. Sua metodologia de propagação e infecção é similar a dos *worms*. *Botnets* são redes formadas por centenas ou milhares de computadores infectados por *bots* e ficam à disposição dos atacantes que as controlam. É comum que *botnets* sejam alugadas para outras pessoas ou grupos que desejam realizar alguma atividade maliciosa específica.
- ***Backdoors:*** são programas cuja finalidade é permitir o retorno do atacante a um computador já comprometido sem a necessidade de recorrer aos métodos de invasão utilizados previamente. Usualmente, a instalação de um *backdoor* consiste na criação de

um novo serviço ou na substituição de determinado serviço por uma versão modificada.

- ***Rootkits:*** são programas que têm como objetivo esconder e assegurar a presença de um invasor ou outro código malicioso em um computador prometido. Algumas funcionalidades de um *rootkit* são, por exemplo, instalação de *backdoors*, remoção de evidências em arquivos de *logs*, ferramentas para a realização de prospecção de redes, entre outras.

2.3 Sistema de detecção de intrusão

Um sistema de detecção de intrusão (IDS) é responsável por monitorar um ambiente, seja um computador ou uma rede de computadores, a fim de detectar intrusos realizando ações maliciosas que possam prejudicar de alguma forma o ambiente monitorado. Tais ações podem ser tentativas ou ataques bem sucedidos que violem a integridade, a confiabilidade ou a disponibilidade de recursos, sejam estes informações ou serviços em execução (ELSHOUSH; OSMAN, 2011).

Cada IDS possui um método específico para realizar a detecção de eventos que ocorram no ambiente monitorado, porém é possível classificá-los em dois principais grupos no que diz respeito à sua técnica de detecção:

- **Detecção por abuso**: envolve os sistemas de detecção que realizam sua detecção com base em regras bem definidas que descrevem as atividades maliciosas. Tais regras são conhecidas como assinaturas de ataque e detalham o padrão de um ataque previamente conhecido, possibilitando o reconhecimento imediato de qual ação está sendo detectada.
- **Detecção por anomalia**: contempla os sistemas de detecção que realizam sua detecção através de distúrbios comportamentais ocorridos no ambiente monitorado. Para realizar a detecção, o IDS considera modelos do comportamento dos usuários e/ou aplicações em execução e considera que desvios presente nestes padrões são

anomalias e possivelmente ações maliciosas. Geralmente este tipo de detecção não possui uma facilidade em reconhecer de imediato qual ação maliciosa está sendo detectada.

Além da classificação quanto ao método de detecção, um IDS também pode ser classificado de acordo com o seu ambiente de monitoramento, podendo ele ser baseado em *host*, em rede ou híbrido, quanto ao foco de detecção, podendo ser centralizado ou distribuído, e em relação à sua capacidade de resposta a um incidente, podendo ele ser passivo ou ativo (WU; BANZHAF, 2010).

Na Figura 2.1 são ilustradas resumidamente as principais classificações que um IDS pode se enquadrar.

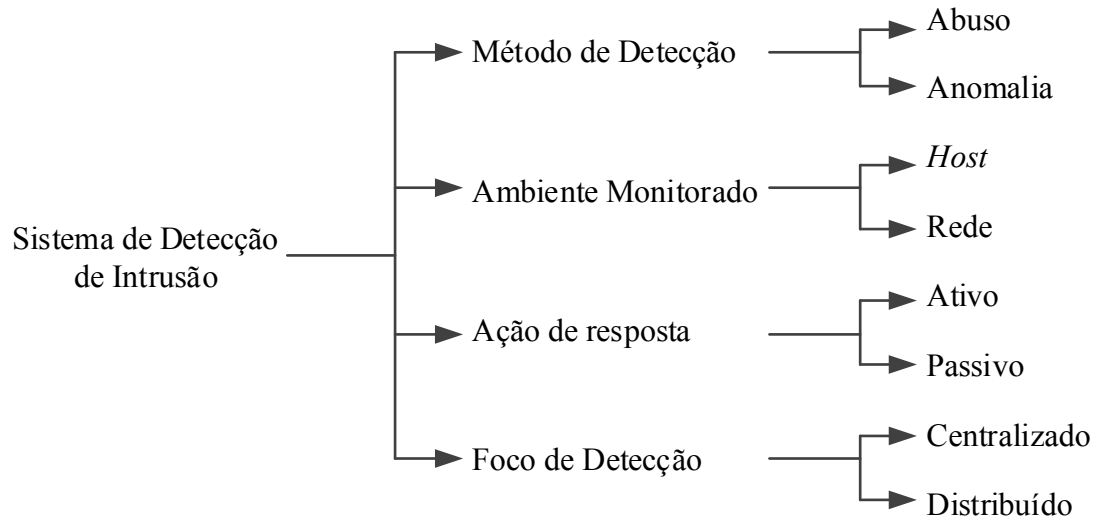


Figura 2.1 Características de um sistema de detecção de intrusão.
Adaptado de (WU; BANZHAF, 2010).

A seguir, é realizada uma descrição no que se refere ao ambiente monitorado:

- **Baseado em *host***: as informações utilizadas são geradas a partir do sistema monitorado, como *logs* dos serviços executados ou chamadas de sistema. Dessa forma, os IDSs atuam de forma a avaliarem apenas aquele determinado *host*. Como vantagem, é possível monitorar

ataques que são provenientes do próprio *host*. Dentre as desvantagens estão o problema com a quantidade de *log* que pode ser gerado por um único sistema, podendo atingir volumes extremamente grandes, além da política da geração desses *logs*.

- **Baseado em rede**: faz uso do próprio tráfego da rede. Todo o tráfego que chegar na interface de rede será capturado. As vantagens desse tipo de sistema é a quantidade de detecção que podem realizar em comparação com sistemas baseados em *hosts*. Como desvantagem é a impossibilidade de avaliar tráfego criptografado. Entretanto, é muito comum que a análise através dessa metodologia não verifica o conteúdo do tráfego.

Em relação aos IDSs que utilização informações provenientes das redes de computadores, há aqueles que fazem a análise do conteúdo dos pacotes que transitam pela rede, enquanto outros analisam apenas as informações provenientes das tentativas de conexões e conexões que são estabelecidas. A técnica de análise do conteúdo dos pacotes, ou *deep packet inspection* (DPI), possibilita a detecção de uma gama muito maior de eventos já que se consegue extrair até as informações referentes à camada de aplicação, enquanto a análise de conexões em sua maioria se limita a analisar as informações até a camada de transporte.

Existem algumas ferramentas *open-source* bem conhecidas que realizam a análise de redes de computadores como o *Snort* (ROESCH, 1999), o *Suricata* (“Suricata”, 2014), que é bastante similar ao *Snort*, porém desenvolvido utilizando técnicas de paralelismo para ser mais escalável, e o *Bro* (PAXSON, 1999). Já para a análise de informações coletadas em *hosts*, há o OSSEC (“Open Source SECurity”, 2014), que realiza análise de *logs*, checa a integridade de arquivos, realiza detecção de *rootkits*, além de outras funcionalidades.

2.4 Fluxo de dados em redes de computadores

Informações sobre tráfego e comportamento de rede são utilizadas por muitas aplicações. Estas informações podem ser extraídas utilizando diferentes equipamentos posicionados estrategicamente dentro da topologia do ambiente analisado, além da utilização de técnicas de análise de *logs* nos *gateways* ou captura de pacotes em determinados pontos da topologia (CORRÊA, 2009).

Com o passar dor tempo, diferentes modelos de exportação de informações foram sendo definidos e desenvolvidos, como o padrão *NetFlow,* descrito no RFC 3954, que foi criado pela *Cisco Systems* e está atualmente na sua versão 9, e o *J-Flow* desenvolvido pela *Juniper*. Entretanto, cada modelo possui suas próprias características de exportação de dados e funcionamento, dificultado a integração destes dados em um sistema de monitoramento.

Devido a esta necessidade, o *Internet Engineering Task Force* (IETF), órgão regulador de padrões da Internet propôs a criação de um padrão para a obtenção e análise de tráfego, conhecido como *IP Flow Information Export* (IPFIX), e está descrito no RFC 7011 (CLAISE; TRAMMELL; AITKEN, 2013).

2.4.1 Padrão *NetFlow*

Um fluxo *NetFlow* é definido como uma sequência unidirecional de pacotes que passam em um dispositivo de rede e possuem algumas propriedades em comum (CLAISE; TRAMMELL; AITKEN, 2013). Um fluxo é uma sumarização do tráfego que possuem as seguintes informações em comum: endereço IP de origem e de destino; porta de origem e de destino (referente ao protocolo da camada de transporte); valor do campo *Protocol* do datagrama IP; byte *Type of Service* do datagrama IP; e interface lógica de entrada do datagrama no dispositivo de rede. O *NetFlow* define ainda a exportação de outras informações sumarizadas além das mencionadas

anteriormente, os campos que são de interesse deste trabalho são apresentados na tabela.

A partir de um ponto da rede é possível posicionar um exportador que irá atuar na sumarização de tais informações e exportar estes dados aos dispositivos coletores onde serão armazenados e constituirão uma valiosa fonte de informações a serem analisadas pelos sistemas de detecção.

2.4.2 Fluxos bidirecionais

Um dos problemas no monitoramento de redes de grande porte é a quantidade de dados gerados para análise, mesmo quando os fluxos de rede *NetFlow* são utilizados. Portanto, o processamento das informações coletadas é algo computacionalmente custoso em termos de tempo e processamento (BATISTA, 2012). Visto que utilizando o padrão de fluxos unidirecionais que o *NetFlow* adota, são gerados dois fluxos distintos para cada conexão, sendo um para cada sentido da conexão. Uma vez que, geralmente, precisa-se da informação completa sobre a conexão, ambos os fluxos são necessários e, portanto, podem ser agrupados como é sugerido no RFC 5103 (TRAMMELL; BOSCHI, 2008), criando assim um fluxo bidirecional.

Ao considerar uma conexão entre dois *hosts* caracterizada por um fluxo unidirecional, tem-se dois fluxos possuindo os mesmos endereços IP dos *hosts*, distinguindo apenas pela especificação da origem e do destino. O que o padrão intitulado *Biflow* faz é unir em um único registro ambos os fluxos pertencentes a uma conexão. Na Figura 2.2 é ilustrada a junção de dois fluxos unidirecionais em um fluxo bidirecional.

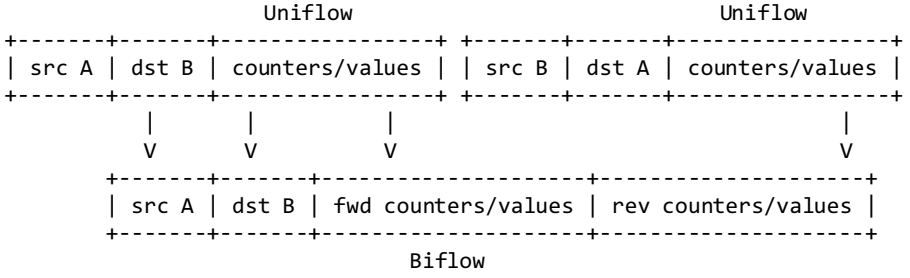


Figura 2.2 Fluxo bidirecional a partir de fluxos unidirecionais (TRAMMELL; BOSCHI, 2008).

Apesar de não ser possível representar todas as conexões de forma bidirecional uma vez que nem sempre é possível determinar os dois lados de uma conexão, ainda assim pode-se obter uma redução significativa na quantidade de informações a serem armazenadas e posteriormente analisadas.

2.5 Aprendizagem de máquina aplicada à detecção de anomalias

Detecção de anomalias é a identificação da ocorrência de eventos inesperados, ou anômalos, dentro de um conjunto de eventos analisados. Dessa forma, é possível destacar atividades que diferem do comportamento normal esperado.

No âmbito de segurança de redes e computadores, a detecção de anomalias possibilita a detecção de eventos sem a necessidade de uma descrição detalhada previamente conhecida. Dessa forma, é possível realizar a detecção de eventos, ou atividades maliciosas, até então não conhecidas.

A detecção de anomalia segue a premissa de que um comportamento anômalo é raro e diferente do comportamento normal. Portanto, o comportamento normal é modelado e todo evento que se desviar de tal comportamento é considerado um evento anômalo. Ainda, é possível classificar os modelos de detecção de anomalia como estático ou dinâmico (CHEBROLU; ABRAHAM; THOMAS, 2005). Os modelos estáticos consideram que o comportamento normal do ambiente monitorado é invariável. Já os modelos dinâmicos assumem que o comportamento normal do ambiente pode sofrer alteração ao longo do tempo, considerando características com os hábitos comportamentais dos usuários e variação no tráfego de rede.

Existem na literatura modelos propostos utilizando variadas técnicas de inteligência artificial, tais como redes neurais artificiais, conjuntos difusos, computação evolutiva, sistemas imunes artificiais e inteligência de enxames (LIAO et al., 2013; WU; BANZHAF, 2010). Logo, é evidente que não um

modelo de detecção que atenda todas as necessidades (MARKOU; SINGH, 2003).

- **Redes neurais artificiais:** este tipo de técnica consiste em uma coleção de unidades de processamento, ou neurônios, que são conectados de acordo com uma topologia. As redes neurais possuem a habilidades de aprender (fase de treinamento) determinados padrões com base em exemplos, tendo a capacidade de classificar novos conjuntos de dados.
- **Conjuntos difusos:** ao contrário de modelos precisos, este tipo de técnica consiste em modelar um problema de forma imprecisa. A importância desta técnica na área de detecção de intruso deve-se ao fato de que os modelos convencionais construídos para detectar intrusões são baseados em atributos numéricos e estatísticas, e considerar uso diretamente de dados numéricos produz um alto índice de erros.
- **Computação evolutiva:** metodologia que toma como base os mecanismos evolutivos presentes na natureza, capaz de resolver problemas de alta complexidade. Por meio dessa classe de algoritmo, é possível resolver problemas que envolvam aleatoriedade, sistemas dinâmicos não lineares, funções multimodais, que são difíceis de serem tratados por algoritmos tradicionais. Duas classes de algoritmos importantes que se enquadram em computação evolutiva (CE) são os algoritmos genéticos e programação genética.
- **Sistemas imunes artificiais:** são aplicados os princípios de imunologia presente no corpo humano em sistemas de detecção de intruso. Este tipo de metodologia funciona com a modelagem de comportamento anômalo dado o comportamento normal.
- **Inteligência de enxames:** compreende técnicas de inteligência artificial envolvendo o estudo do comportamento coletivo em sistemas descentralizados. O processo ocorre através da emulação do comportamento de insetos ou enxames de abelha de modo a simplificar a concepção de soluções distribuídas para problemas complexos.

Os algoritmos de aprendizagem de máquina podem ser divididos em supervisionados e não supervisionados. Os algoritmos supervisionados aprendem através do treinamento utilizando dados rotulados, isto é, dados previamente classificados, seja por um humano ou outro sistema especialista. Já os algoritmos não supervisionados não requerem dados rotulados durante sua etapa de treinamento. Embora os algoritmos supervisionados, em geral, sejam mais precisos, conseguir dados rotulados é uma tarefa difícil na prática, seja por dificuldade em rotular manualmente o conjunto de treinamento ou por problemas relacionados à privacidade (CRETU et al., 2008). Portanto, o uso de métodos não supervisionados ajudam a lidar com este problema uma vez que não necessitam de dados rotulados em seu treinamento.

Os algoritmos de agrupamento e redes neurais artificiais são frequentemente utilizados em sistemas de detecção de intrusão não supervisionados. Ainda, os métodos não supervisionados possibilitam a identificação de anomalias presentes nos dados, em geral após a definição de um limiar, assumindo que os dados normais são comuns enquanto as anomalias são eventos raros presentes em regiões esparsas do espaço de características (ESKIN et al., 2002).

2.5.1 Algoritmos de agrupamento

Os algoritmos de agrupamento são métodos de aprendizagem de máquina não supervisionados que possibilitam a identificação de conjuntos de dados que possuem características semelhantes (GENTLEMAN; CAREY, 2008). Os dados são divididos em grupos, conhecidos também como classes, a partir de suas características similares.

Através do uso de algoritmos de agrupamento, é possível identificar as classes de um problema mesmo que estas não sejam conhecidas *a priori* (ZANDER; NGUYEN; ARMITAGE, 2005). Dessa forma, estes algoritmos são indicados para a classificação de tráfego de rede visto que as aplicações

possuem comportamentos dinâmicos e geram diversas classes (NGUYEN; ARMITAGE, 2008; ZANDER; NGUYEN; ARMITAGE, 2005).

Diferentes métodos de agrupamento podem ser encontrados e não há um método que seja a solução definitiva para todos os problemas, isto porque existem abordagens distintas para lidar com diferentes conjuntos de dados. As abordagens mais comuns são (GAMA, 2010):

- **Métodos de particionamento:** o conjunto de dados é particionado em k grupos, minimizando uma função objetivo. Em geral são utilizados para encontrar grupos convexos.
- **Métodos que utilizam micro-grupos:** dividem o processo de agrupamento em duas etapas, uma online e outra off-line. Na etapa online, os dados são sumarizados em modelos locais (micro-grupos). Já na segunda etapa, o modelo de agrupamento global é gerado. Podem ser utilizados para agrupar fluxos contínuos de dados (*streaming*).
- **Métodos baseados em densidade:** se baseiam em conectividade entre regiões e funções de densidade. Os grupos são formados a partir de regiões densas de objetos no espaço de dados. São utilizados para encontrar grupos que possuem formatos arbitrários.
- **Métodos baseados em grades:** os objetos são divididos em células que formam uma estrutura de grade multidimensional. Cada célula tem sua densidade avaliada e células conectadas lateralmente são consideradas adjacentes. A vantagem destes métodos é que sua complexidade está atrelada ao número de células da grade e não ao número de pontos, possibilitando analisar grandes volumes de dados.

2.6 Considerações finais

Neste capítulo foram abordados os conceitos necessários para o entendimento deste trabalho. Foram apresentados alguns ataques comuns a

redes de computadores, conceitos sobre fluxos de dados em redes de computadores, noções sobre os sistemas de detecção de intrusão e suas características, além das técnicas de aprendizado de máquina utilizadas na detecção de anomalias.

Dentre as diferentes classificações nas quais um IDS pode ser designado, a detecção por anomalias possibilita a detecção de eventos até então não conhecidos. A detecção de anomalias pode ser obtida através de diferentes técnicas, como a utilização métodos estatísticos, redes neurais artificiais, sistemas imunes artificiais, dentre outras técnicas de aprendizado de máquina. Algumas técnicas exigem um maior poder de processamento que outras, portanto, na análise de grandes volumes de dados, tal como uma rede de computadores, este é um aspecto importante durante a escolha do método utilizado. No capítulo seguinte são apresentados os trabalhos relacionados ao projeto proposto.

CAPÍTULO 3 - Trabalhos relacionados

3.1 Considerações iniciais

Neste capítulo são apresentados os trabalhos relacionados ao projeto proposto. Embora existam diversos trabalhos relacionados ao tema detecção de intrusão, são apresentados aqui os trabalhos encontrados relacionados à detecção de anomalias, além da utilização de técnicas de aprendizado de máquina.

A detecção de intrusão em redes de computadores é tema de pesquisas desde 1987 (DENNING, 1987) até os dias atuais (AHMED; NASER MAHMOOD; HU, 2016). Os sistemas de detecção de intrusão podem ser classificados de acordo com o seu método de detecção: detecção por abuso e detecção de anomalias. Visto que as técnicas de detecção por abuso são capazes de detectar apenas ataques previamente conhecidos, a área de detecção de anomalias tem sido mais ativa recentemente (CHEN et al., 2014). Uma extensa revisão sobre a detecção de anomalias é apresentada por Chandola et al. (2009). Ahmed et al. (2016) apresentam uma revisão na área de detecção de anomalias especificamente em redes de computadores.

Neste capítulo é realizado um levantamento dos trabalhos relacionados ao projeto proposto. Embora existam diversos trabalhos relacionados ao tema detecção de intrusão, são apresentados aqui os trabalhos encontrados relacionados à detecção de anomalias que fazem uso de técnicas de

aprendizado de máquina, realizam análise de fluxos contínuos de dados ou abordam sistemas adaptativos.

3.2 Detecção de intrusão utilizando aprendizado de máquina

Dada a complexidade para a geração de modelos capazes de descreverem o comportamento normal de um ambiente, técnicas de aprendizado de máquina têm sido amplamente utilizadas. Estes métodos podem ser classificados como supervisionados ou não supervisionados.

Métodos supervisionados necessitam de dados rotulados para a execução de seu treinamento e alguns métodos populares são k-*nearest neighbor* (k-NN), *naive* Bayes, árvores de decisão, *support vector machines* (SVM) e redes neurais artificiais. Há também métodos baseados em sistema imune artificial (AIS) inspirados no sistema imunológico humano.

Uma revisão geral sobre os sistemas de detecção de intrusão é apresentada em (LIAO et al., 2013) e uma revisão detalhada sobre métodos supervisionados e não supervisionados aplicados à detecção de intrusão é apresentada por Wu e Banzhaf (2010).

3.2.1 Detecção de anomalias utilizando métodos não supervisionados

Embora os métodos supervisionados sejam bastante precisos, a dificuldade em obter conjuntos abrangentes de informações rotuladas para o treinamento destes impõe algumas limitações, especialmente quando aplicados à detecção de ataques a redes de computadores. Estes ataques são facilmente modificados de modo a evadir os padrões previamente conhecidos, além da possibilidade de ocorrência de ataques ainda não conhecidos.

Portnoy et al. (PORTNOY; ESKIN; STOLFO, 2001) apresentaram uma metodologia de detecção de intrusão utilizando o método de agrupamento hierárquico e utilizaram o conjunto de dados KDD'99 para avaliar seu

desempenho. Uma comparação entre o uso de técnicas de agrupamento, k-NN e one-class SVM é apresentada em (ESKIN et al., 2002), os autores compararam o desempenho dos diferentes métodos também utilizando o conjunto de dados KDD'99. ZHONG; KHOSHGOFTAAR; SELIYA (2007) compararam os métodos k-médias, mixture-of-spherical Gaussians, redes neurais SOM (*self-organizing maps*) e *neural-gas*, com o k-médias sendo o mais rápido e sua versão online a mais precisa.

No trabalho apresentado por Lin, Ke e Tsai (2015) é proposta uma nova maneira de representar as características dos dados. No método proposto, nomeado *CANN* (*Cluster center And Nearest Neighbor*), a distância entre cada objeto e o centroide de seu grupo é somada com a distância entre o objeto e seus vizinhos mais próximos no mesmo grupo. Cada objeto é representado por esta nova característica unidimensional e um classificador k-NN é utilizado para a detecção de intrusos. Costa et al. (2012) apresentaram em seu trabalho a aplicação do *OPFC*, um método de agrupamento não supervisionado baseado no algoritmo *OPF* (*Optmum-Path Forest*), no problema de detecção de intrusão. O algoritmo *OPFC* foi comparado com outros dois algoritmos não supervisionados, k-médias e SOM, onde o algoritmo OPFC apresentou uma melhor separação dos dados em quatro dos oito conjuntos de dados testados.

Ainda considerando o uso de métodos de agrupamento, uma abordagem baseada em uma técnica de agrupamento em subespaços baseada em árvore, nomeada *TreeCLUS*, é apresentada em (BHUYAN; BHATTACHARYYA; KALITA, 2012). O algoritmo *TreeCLUS* emprega conceitos da teoria da informação, utilizando a técnica MMIFS (AMIRI et al., 2011), para identificar o subconjunto de características relevantes. É realizada uma análise de estabilidade do algoritmo de agrupamento utilizando um conjunto de medidas para avaliar sua eficiência em encontrar todos os grupos possíveis. As medidas utilizadas são os índices *Dunn*, C-index, Davies Bouldin, Xie-Beni e o coeficiente de silhueta. Finalmente, os grupos resultantes são rotulados utilizando o algoritmo *CLUSLab*, que emprega um conjunto de técnicas e rotula as instâncias de um grupo de acordo com o tamanho do grupo, sua compacidade e seu conjunto de características

predominantes. Os algoritmos propostos são avaliados utilizando os conjuntos de dados KDD'99 e TUIDS. A fim de obter mais informações sobre detecção não supervisionada de anomalias, é apresentada uma visão geral recente do uso de algoritmos de agrupamento em (BHUYAN; BHATTACHARYYA; KALITA, 2014).

3.3 Sistemas de detecção adaptativos

Os trabalhos apresentados até então têm como objetivo colaborar com a área de detecção de intrusos através do uso de algoritmos de aprendizado de máquina não supervisionados. Entretanto, não abordam a dificuldade na adaptação do modelo de detecção. Nesta seção são apresentados alguns trabalhos voltados a área de detecção e identificação de anomalias que apresentam esforços para realizar a detecção de maneira online e adaptativa.

Spinosa e outros (SPINOSA; DE LEON F. DE CARVALHO; GAMA, 2008) propõem uma nova abordagem para a detecção de novidades em fluxos contínuos de dados. A abordagem proposta, *OLINDDA* (*OnLIne Novelty and Drift Detection Algorithm*), é dividida em duas fases: uma fase supervisionada, onde o modelo normal é construído a partir de um conjunto de exemplos que descrevem o domínio de dados, e uma fase de não supervisionada, onde os elementos analisados são processados e novos conceitos são aprendidos continuamente. O método proposto foi avaliado utilizando o conjunto de dados KDD'99 e foi capaz de encontrar classes de ataques como novos conceitos.

Em (HU; REN; REN, 2009) é proposto um algoritmo de detecção de anomalias baseado em agrupamento hierárquico. O algoritmo proposto, nomeado ADBHC, gera os grupos utilizando particionamento baseado em densidade e utiliza uma árvore de agrupamento hierárquico para realizar a detecção adaptativa das anomalias. Os perfis normais são gerados, representados por uma árvore de grupos, e aplicado um algoritmo de poda baseado na densidade dos nós. Então, cada registro de conexão de rede é comparado com os perfis normais e, se o registro não se encaixa em nenhum

grupo da árvore, um alerta é emitido. Os perfis normais são atualizados a partir dos registros de conexões normais.

No trabalho (THAKRAN; TOSHNIWAL, 2012), é proposto um método para detecção de anomalias em fluxo continuo de dados combinando os algoritmos DBSCAN e k-médias ponderado. O fluxo é analisado em porções de dados. Cada porção de dados é agrupada utilizando o algoritmo DBSCAN, onde os grupos pequenos e os *outliers* são considerados candidatos a anomalia e são analisados novamente, junto com as próximas porções de dados, até que sua pontuação atinjam um limiar L. A porção de dados é então analisada utilizando algoritmo k-médias ponderado, que atribui pesos aos atributos de acordo com sua relevância no agrupamento. Os grupos obtidos servem de entrada para o módulo de detecção de anomalias. Os candidatos a anomalias são analisados diversas vezes até que sejam considerados anomalias ou sejam descartados. O método proposto foi testado em conjuntos de dados sintéticos e conjuntos reais retirados do repositório UCI (LICHMAN, 2013) e os resultados foram comparados com o algoritmo *CORM* (*Cluster based OutlieR Miner*), apresentando uma taxa de detecção superior e uma taxa de falsos positivos mais baixa.

Em (WANG et al., 2014), os autores propuseram um *framework* de detecção de intrusão autônomo capaz de se adaptar à detecção de fluxo de tráfego HTTP. Enquanto a maioria dos trabalhos na literatura se preocupam com a detecção de anomalias em conjuntos de dados previamente coletados, este trabalho leva em consideração as características dos fluxos contínuos de dados analisados em ambientes de produção, possibilitanto, assim, uma detecção quase em tempo real. A abordagem proposta possui três pontos de destaque: sua capacidade de identificando anomalias em fluxos de dados não rotulados; atualização do modelo de detecção, incorporando dados rotulados como normais; e adaptação do modelo de detecção após uma mudança no comportamento do fluxo de dados. Esta abordagem utiliza o algoritmo de clusterização AP (*Affinity Propagation*) (FREY; DUECK, 2007) e sua extensão para *streaming* de dados (ZHANG; FURTLEHNER; SEBAG, 2008) para a detecção de comportamentos anômalos. Os experimentos utilizando o algoritmo de clusterização AP foram realizados utilizando o conjunto de dados

KDD'99, comparados com as técnicas *k*-NN *(k-Nearest Neighbor)*, PCA *(Principal Component Analysis)*, SVM *(Suport Vector Machine)* e SKL *(Sequential Karhunen–Loeve)* e apresentaram resultados satisfatórios. Na Figura 3.1 são apresentadas as curvas ROC representando o desempenho dos algoritmos em relação à taxa de verdadeiros positivos e falsos positivos.

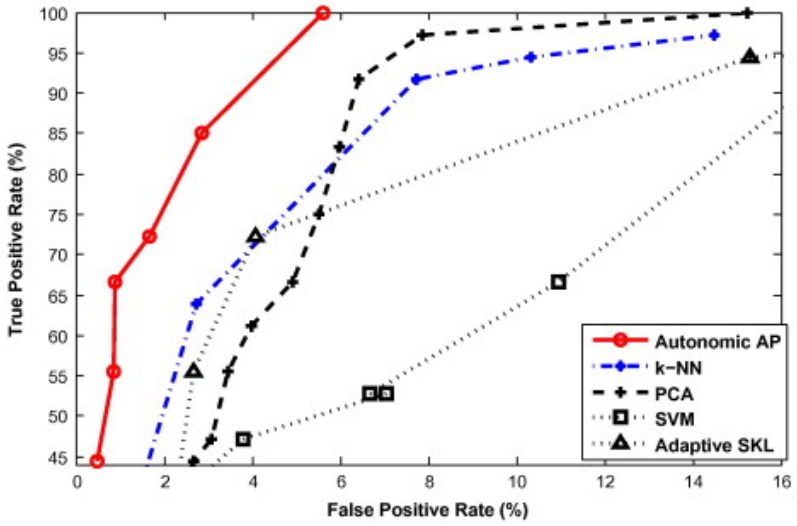


Figura 3.1. Comparação entre a detecção de ataques utilizando os algoritmos AP, *k*-NN, PCA, SVM e SKL.

3.4 Detecção em tráfego de rede e grandes volumes de dados

Esta seção apresenta outros trabalhos que também contribuíram para o desenvolvimento do presente projeto.

Hsiao e outros (2010) apresentam uma arquitetura para um sistema de detecção de sites maliciosos através da análise de tráfego de rede. O modelo de detecção é construído a partir de fluxos *NetFlow* utilizando um método espacial-temporal de agregação de variáveis. A partir dos dados originalmente presentes no padrão *NetFlow*, são derivadas outras características. A partir da análise dos resultados é possível concluir que tais variáveis obtiveram um melhor resultado em relação às variáveis originais do protocolo *NetFlow* quando aplicadas às técnicas de detecção

No trabalho apresentado por Suthaharan (2013), são apresentados os desafios em relação à detecção de intrusão em *Big Data* utilizando aprendizado de máquina. O autor ressalta as características de *Big Data*, sendo elas volume, variedade e velocidade (ZIKOPOULOS; EATON, 2011), e mostra que os dados de tráfego de rede satisfazem as características para serem classificados como *Big Data.* Ainda, é feita a proposta de uma nova definição para *Big Data* introduzindo um espaço tridimensional, C^3, que é definido com base nos parâmetros de cardinalidade, continuidade e complexidade, facilitando a utilização de ferramentas matemáticas e estatísticas.

Os principais desafios apresentados em relação ao gerenciamento do grande volume de dados estão relacionados à topologia da rede, comunicação e segurança. Já em relação aos algoritmos para a análise desses dados, deve-se levar em consideração as dimensões do conjunto de dados, a grande quantidade de tipos de dados, a alta velocidade na qual esses dados devem ser processados e os dados não estruturados. Como conclusão, o autor sugere a integração de tecnologias existentes para o tratamento de *Big Data*, como a utilização do HDFS (*Hadoop Distributed File System)* e a computação em nuvem para o processamento dos dados, aliadas a técnicas de aprendizado de máquina aprimoradas para este contexto.

3.5 Considerações finais

Neste capítulo foram apresentados os trabalhos relacionados à detecção de eventos anômalos utilizando técnicas de análise de grandes volumes de dados. Esses artigos foram utilizados como base para o desenvolvimento da proposta deste trabalho a ser apresentada no próximo capítulo.

Este trabalho de difere dos trabalhos apresentados em alguns aspectos como: a) uma abordagem para a detecção de anomalias em fluxo contínuo de dados de uma maneira totalmente não supervisionada, eliminando a necessidade uma etapa de treinamento utilizando um conjunto de dados

rotulados; b) adaptação continua do modelo com uma baixa complexidade computacional; e c) combinação de duas técnicas de agrupamento de dados para a detecção de anomalias em redes de computadores através da análise de fluxo de rede.

CAPÍTULO 4 - Metodologia

4.1 Considerações iniciais

Neste capítulo é descrita a metodologia utilizada no desenvolvimento deste trabalho. O trabalho proposto consiste em realizar a análise de dados provenientes do tráfego de redes de computadores, com o objetivo de detectar eventos anômalos que possam ocorrer no ambiente monitorado. A metodologia proposta é uma combinação de técnicas de agrupamento que não exige uma fase de treinamento off-line e tem a capacidade de se adaptar a mudanças no comportamento de rede.

Na seção 4.2 são apresentados os objetivos deste trabalho. Seguindo, na seção 4.3 é apresentada a arquitetura do sistema proposto por este trabalho, descrevendo como os diferentes módulos são integrados. Na seção 4.4 é apresentada a descrição das informações extraídas dos fluxos de rede, bem como a especificação das caraterísticas utilizadas para a criação do modelo de detecção proposto. Na seção 4.5 é detalhada a identificação de fluxos suspeitos utilizando o algoritmo DBSCAN. Na seção 4.6 é apresentada a construção e atualização do modelo de detecção. Na seção 4.7 são apresentados os critérios utilizados para a detecção das anomalias. Concluindo, na seção 4.8 são feitas as considerações finais sobre este capítulo.

4.2 Objetivos

O projeto proposto tem como objetivo primário a elaboração de um modelo de detecção de anomalias em redes de computadores utilizando uma combinação de métodos de aprendizagem não-supervisionados capaz de analisar grandes volumes de dados e se adaptar a mudanças de comportamento através de um aprendizado constante.

Dentre os métodos de aprendizagem de máquina não-supervisionados, são combinados os algoritmos de agrupamento DBSCAN e k-médias que, dado um conjunto de dados de entrada, possuem o objetivo de agrupar os dados semelhantes. O objetivo do uso de algoritmos de agrupamento é encontrar a separação entre tráfego normal e tráfego malicioso.

Embora existam diversas pesquisas na área de detecção de intrusão utilizando algoritmos de agrupamento, são poucas as que se preocupam com o desempenho e a capacidade de adaptação do modelo a mudanças de comportamento (FENG et al., 2014). Uma abordagem para processar grandes volumes de dados é considerar os problemas presentes no processamento de Fluxo Contínuo de Dados (FCD), onde o algoritmo utilizado deve ser rápido, incremental e lidar com dados infinitos e não estacionários (SILVA et al., 2013). Através do método proposto foram alcançadas características que satisfazem as necessidades para o tratamento de fluxos contínuos.

4.3 Arquitetura proposta

Um fluxo contínuo de dados DS é uma sequência de elementos $x_1, x_2, \dots, x_m$, possivelmente infinita, que pode possuir variações em sua distribuição. Cada elemento x_i é caracterizado por um conjunto de n atributos, isto é, $x_i = (x_i^1, x_i^2, \dots, x_i^n)$.

Neste trabalho os dados são analisados em porções. Cada porção contém m objetos que são processados em conjunto. A quantidade de dados processados em cada porção pode ser especificada pelo usuário e depende da natureza dos dados, tal como velocidade de transmissão e tempo necessário para processamento.

A detecção de eventos maliciosos ocorre através de três etapas. Na primeira etapa, a porção de dados é agrupada utilizando o algoritmo de agrupamento baseado em densidade DBSCAN. O resultado da primeira etapa é a divisão dos dados em dois conjuntos, um conjunto composto por dados considerados normais e outro conjunto composto por dados considerados suspeitos. Na segunda etapa, o modelo do algoritmo k-médias é atualizado utilizando o conjunto de dados considerados normais durante a primeira etapa. Na terceira etapa, cada elemento do conjunto de dados suspeitos tem sua distância calculada em relação aos centroides do algoritmo k-médias e, se estiver muito distante, é considerado anômalo. Ainda, os objetos considerados suspeitos, mas que não foram claramente identificados, são encaminhados para uma possível análise externa.

A integração dos módulos que compõem o sistema proposto é ilustrada na Figura 4.1. O módulo que utiliza o algoritmo k-médias é conhecido como modelo global uma vez que este modelo é construído apenas uma vez e então é constantemente atualizado. Dessa maneira, seus grupos são influenciados por todos os dados analisados, enquanto que o módulo que utiliza o algoritmo DBSCAN gera novos grupos a cada porção de dados analisada.

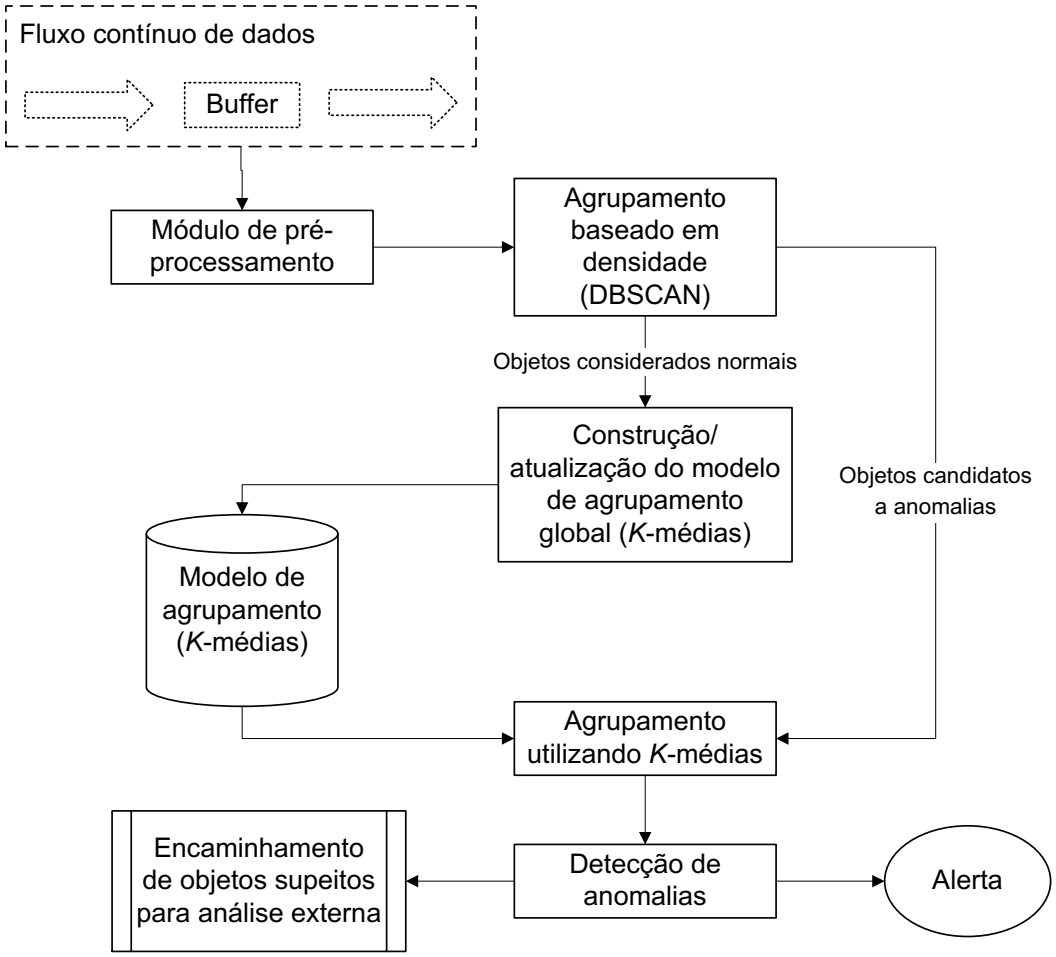


Figura 4.1 Arquitetura do sistema proposto.

4.4 Definição e extração das características

Uma vez realizada a coleta dos fluxos de rede, os dados são pré-processados para a extração de algumas características que servirão de descritores para o módulo de agrupamento. O resumo das informações dos fluxos de rede é realizado considerando endereços IP de origem distintos. Dessa forma, é possível modelar o comportamento de cada usuário ou servidor individualmente, agrupar os que possuem comportamento semelhante e identificar comportamentos anômalos. Na Tabela 4.1 são descritas as características utilizadas como descritores para o módulo de treinamento.

Tabela 4.1 Descritores utilizados para o agrupamento de tráfego.

#ID	Nome	Descrição
1	num_con	Quantidade de fluxos/conexões que um determinado IP de origem realizou.
2	dist_addr	Quantidade de endereços de destino distintos que um determinado IP de origem acessou.
3	dist_port	Quantidade de portas distintas que um determinado IP de origem tentou se conectar.
4	null_rate	Taxa dos fluxos que não obtiveram resposta por parte do destinatário.
5	syn_rate	Taxa dos fluxos que tiveram a *flag* SYN do protocolo TCP ativada.
6	rst_rate	Taxa dos fluxos que tiveram a *flag* RST do protocolo TCP ativada.
7	src_pkts_rate	Taxa de pacotes enviados por conexão.
8	src_octets_rate	Taxa de bytes enviados por conexão.
9	dst_pkts_rate	Taxa de pacotes recebidos por conexão.
10	dst_octets_rate	Taxa de bytes recebidos por conexão.
11	time_rate	Média da duração das conexões.

A fim de visualizar as diferenças e semelhanças entre as características, foram gerados dois mapas de características, onde cada linha apresenta as informações de um endereço de origem distinto, enquanto as colunas representam as informações contidas na Tabela 4.1. Para a geração dos mapas de características os valores obtidos foram normalizados entre 0 e 1, sendo que quanto mais próximo de 0, mais clara é a cor da célula, e quanto mais próximo de 1, mais escura.

Na Figura 4.2 são apresentadas as características referentes a um conjunto de 50 amostras aleatórias sem cunho malicioso. É possível notar que algumas linhas possuem semelhança entre si, tais linhas serão possivelmente agrupadas em um mesmo grupo após a utilização do algoritmo de agrupamento.

Figura 4.2. Mapa de características sem ataque.

Do conjunto de 50 amostras representados na Figura 4.2, a amostra referente à primeira linha foi substituída por uma amostra relativa a uma varredura de serviços em rede. O resultado da inserção desta amostra maliciosa é apresentado no mapa de características da Figura 4.3. Há uma alteração significativa nas características 1, 3 e 6, referentes ao número total de conexões, à quantidade de portas distintas acessadas e à taxa de fluxos que tiveram a *flag* RST do protocolo ativada.

Figura 4.3. Mapa de características, onde a primeira linha representa um ataque.

Tal representação corrobora para a teoria de que as conexões de tráfego normal serão agrupadas, enquanto o tráfego anômalo será agrupado em grupos isolados, ou ainda, serão agrupados em grupos de tráfego normal, mas sua distância em relação ao centro será maior do que os outros elementos do grupo.

4.5 Identificação de fluxos suspeitos utilizando o algoritmo DBSCAN

Cada porção de dados é analisada utilizando o algoritmo DBSCAN. A escolha do algoritmo DBSCAN se deu pelo fato de que este não requer uma

definição prévia do número de grupos, é capaz de encontrar grupos com diferentes formas e, ainda, tem a capacidade de identificar ruídos presentes nos dados. Os ruídos identificados pelo algoritmo DBSCAN são considerados fluxos suspeitos e então enviados para análise no módulo de detecção.

Além dos ruídos encontrados pelo DBSCAN, os grupos obtidos como resultado do agrupamento têm sua densidade relativa avaliada. A densidade relativa de um grupo c_i é calculada conforme (4.1), onde Dc_i é a densidade do grupo c_i e ED é a densidade esperada de todos os grupos encontrados.

$$RDc_i = \frac{Dc_i}{ED} \tag{4.1}$$

A densidade de um grupo D_c pode ser medida de diversas maneiras, como a quantidade de objetos pertencentes ao grupo ou a distância média entre estes objetos. Neste trabalho, foi utilizada a quantidade total de objetos atribuídos ao grupo. Os grupos c_i cuja $RDc_i < \delta$ são considerados anômalos, onde δ é a densidade relativa mínima para que um grupo seja considerado normal. Assume-se que, se um grupo possui poucos elementos em relação ao esperado, este grupo é formado por objetos fora do padrão esperado e, portanto, anômalos.

4.6 Construção e atualização do modelo de detecção

O modelo de detecção utiliza o algoritmo k-médias para definir os grupos de conexões normais e calcular o centroide de cada grupo. Os centroides encontrados são utilizados para na detecção das anomalias, como será explicado na seção 4.7.

A motivação para a escolha do algoritmo k-médias se deu pela baixa complexidade para realizar atualizações no modelo de agrupamento. Como resultado do treinamento do algoritmo k-médias são obtidos os centroides de cada grupo, isto é, vetores que representam os valores médios dos elementos de cada grupo. Logo, para realizar uma atualização em um modelo previamente existente, é possível realizar uma média entre os valores prévios e os centroides dos dados analisados no momento.

A grande dificuldade na utilização do k-médias é a definição da quantidade k grupos, ou *clusters*, utilizados para a realização do agrupamento. Em (PHAM; DIMOV; NGUYEN, 2005) os autores fazem uma discussão sobre diversas formas de definição do valor k encontradas na literatura, expondo o impacto que estas escolhas têm no desempenho do algoritmo, e ressalta a importância da realização de testes utilizando diversos valores para *k* e da proporção deste valor em relação ao tamanho do conjunto de dados.

Os dados considerados normais durante a primeira etapa são utilizados para atualizar o modelo global. Esta abordagem de utilizar os dados considerados normais para atualização do modelo de detecção é baseada nos métodos propostos em (HU; REN; REN, 2009; THAKRAN; TOSHNIWAL, 2012). Após a atribuição dos objetos considerados normais a cada grupo do modelo global, o centroide de cada grupo é atualizado de acordo com

$$c_{t+1} = \frac{c_t n_t \alpha + e_t m_t}{n_t \alpha + m_t} \tag{4.2}$$

$$n_{t+1} = n_t + m_t \tag{4.3}$$

em que c_t é o centroide a ser atualizado no instante t, n_t é a quantidade de objetos associados ao grupo até o momento, e_t é o novo centroide da porção de dados sendo analisada no momento e m_t é a quantidade de objetos adicionados a este grupo no instante t. A constante α define a influência que os dados prévios possuem na atualização e pode possuir valores entre 0 e 1, sendo que, para $\alpha = 1$, todos os dados são utilizados desde o começo e, para $\alpha = 0$, somente os dados da porção atual são considerados para a definição do novo centroide c_{t+1}.

Essa atualização garantirá que o modelo global seja constantemente atualizado de modo que, se houver alguma mudança no comportamento normal, esta mudança será gradativamente incorporada ao modelo.

4.7 Detecção de anomalias

Após a atualização dos centroides, todos os candidatos a anomalias provenientes da primeira etapa são analisados novamente utilizando o modelo de detecção. O objetivo desta segunda análise é tentar identificar apenas os objetos realmente anômalos e minimizar a quantidade de falsos positivos. Durante a primeira etapa, objetos normais podem ser considerados possíveis anomalias por alguns motivos, como fazer parte de um grupo de conexões dividido em duas porções ou mesmo possuir características anormais em relação à porção atual, mas que já foram presenciadas anteriormente.

Após o agrupamento utilizando o modelo global, cada objeto suspeito o_j tem sua distância em relação ao centro c_i calculada. A distância obtida é utilizada para avaliar se um objeto está fora da fronteira estabelecida pelo limiar ε, isto é, se $d(c_i, o_j) > \varepsilon$, o_j é considerado anômalo. Neste trabalho, duas métricas diferentes são utilizadas a fim de comparação. A primeira métrica calculada é a distância euclidiana. A segunda métrica é calculada considerando o desvio baseado em limiar (IPPOLITI; ZHOU, 2012). Neste trabalho, esta distância será referenciada como TEV e o cálculo da TEV é explicado a seguir.

4.7.1 Cálculo da distância TEV

Para cada objeto o_j, é calculado o vetor QE (4.4) com a diferença entre o objeto e o centro para cada atributo k.

$$QE = [qe_j^1, qe_j^2, \dots, qe_j^n] \tag{4.4}$$

em que $qe_j^k = \left|c_i^k - o_j^k\right|, k = 1,2, \dots, n.$

O valor de TEV_j é calculado a partir da soma dos desvios que ultrapassem o limiar τ, conforme:

$$TEV_j = \sum_{k=1}^{n} f(k) \tag{4.5}$$

$$f(k) = \begin{cases} qe_j^k & se\ qe_j^k > \tau_1 \\ 0 & se\ qe_j^k \leq \tau_1 \end{cases} \quad (4.6)$$

O parâmetro τ_1 controla quão próximo um atributo do objeto analisado deve estar em relação ao mesmo atributo do centroide de modo que este atributo não seja levado em consideração durante o cálculo da distância TEV.

Uma conexão anômala geralmente possui atributos com valores muito próximos dos atributos de uma conexão normal, com exceção de alguns deles (IPPOLITI; ZHOU, 2012). A ideia de utilizar a distância TEV é justamente considerar apenas os atributos que diferem do centro do grupo significativamente, enquanto que a distância euclidiana leva em consideração a diferença entre todos os atributos.

4.7.2 Encaminhamento de conexões suspeitas

O método proposto neste trabalho realiza uma análise apenas das informações sobre o fluxo de rede, isto é, não considera o conteúdo dos pacotes. Dessa forma, ataques que possuem tráfego de rede similar a conexões legítimas são difíceis de serem detectados, exigindo uma inspeção profunda de pacotes (DPI – *Deep Packet Inspection*). A fim de possibilitar a detecção de ataques que seriam detectados apenas utilizando técnicas de DPI, foi acoplado à metodologia um mecanismo de encaminhamento de tráfego que apresentam um comportamento suspeito, mas que não foram detectados como anomalias pelo módulo de detecção.

Os dados são encaminhados a um IDS que realize então a análise de conteúdo, por exemplo, *Snort*. Neste trabalho não foram avaliados os sistemas de detecção que realizam DPI. Assume-se que estes não possuem casos de falsos positivos uma vez que possuem assinaturas específicas para os ataques detectados. O que pode ocorrer, no entanto, é que estes não possuam assinaturas para detectar o ataque em questão, gerando um falso negativo. As conexões são encaminhadas para análise externa caso o valor da distância calculada esteja entre τ_2 e ε, (4.7).

$$\tau_2 \leq d(c_i, o_j) \leq \varepsilon \quad (4.7)$$

Na Figura 4.4 é ilustrado um grupo encontrado em um conjunto bidimensional contendo alguns objetos normais, suspeitos e anômalos.

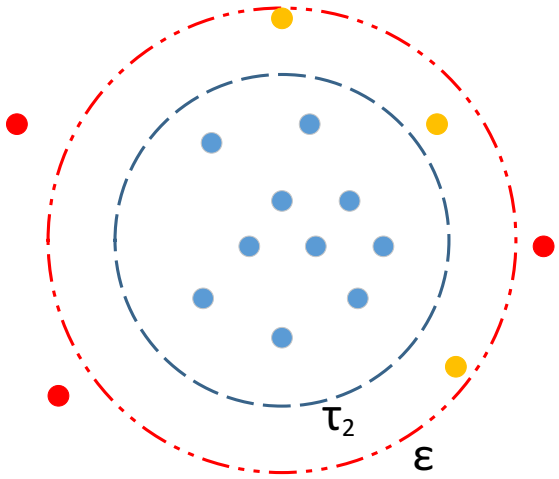


Figura 4.4 Representação de um grupo: objetos considerados normais (azul), objetos suspeitos encaminhados (amarelo) e objetos anômalos (vermelho).

4.8 Considerações finais

Neste capítulo foi realizada uma descrição da metodologia utilizada, bem como a arquitetura do sistema de detecção proposto, o processo de extração de características, os critérios para identificação de anomalias e o mecanismo de encaminhamento de fluxos suspeitos. A metodologia proposta utiliza duas técnicas de agrupamento distintas, além de utilizar uma estratégia de encaminhamento de objetos suspeitos que não possuem um alto nível de certeza quanto a sua natureza.

No próximo capítulo são descritos os experimentos realizados, descrevendo os conjuntos de dados e os resultados obtidos comparando com variações diferentes da metodologia proposta.

CAPÍTULO 5 - Experimentos e resultados

5.1 Considerações iniciais

Neste capítulo são apresentados os experimentos realizados e os resultados obtidos a partir da abordagem proposta nesta pesquisa. A fim de validar método proposto, foram realizados testes utilizando dois conjuntos de dados distintos contendo informações sobre conexões normais e conexões maliciosas. Na seção 5.2 são apresentadas as métricas utilizadas para realizar a análise dos resultados. Na seção 5.3 são apresentados os resultados obtidos a partir dos experimentos utilizando o conjunto de dados NSL-KDD. Na seção 5.4 é descrito o conjunto de dados ACME-IDE. Já na seção 5.5 são descritos os resultados obtidos a partir da análise do conjunto ACME-IDE. Por fim, na seção 5.6 encontram-se as considerações finais sobre os resultados obtidos e o desempenho da metodologia proposta.

5.2 Métricas utilizadas para análise dos resultados

Alguns trabalhos que utilizam técnicas de agrupamento apresentam os resultados obtidos em relação à pureza dos grupos. Esta é uma métrica bastante propícia para avaliar se objetos de uma mesma classe foram agrupados dentro do mesmo grupo. Entretanto, a metodologia proposta tem como objetivo ser um classificador binário, seguindo a premissa de que todos

os grupos representam objetos normais e as anomalias são objetos estão distantes destes grupos. A classificação dos dados normais está fora do escopo deste trabalho.

O desempenho do método proposto foi mensurado por meio de métricas empregadas na análise de classificadores binários, como é o caso da detecção de anomalias (WU; BANZHAF, 2010). Quando uma anomalia é detectada corretamente, têm-se um verdadeiro positivo (TP), caso contrário, têm-se um falso negativo (FN). Um verdadeiro negativo (TN) é um caso onde um fluxo normal é classificado corretamente, caso este fluxo seja identificado como anomalia, têm-se um falso positivo (FP). A matriz de confusão apresentada na Tabela 5.1 ilustra os casos descritos anteriormente.

Tabela 5.1 Matriz de confusão.

		Valor Predito	
		Positivo (Anomalia)	Negativo (Normal)
Valor real	Positivo (Anomalia)	**TP**	**FN**
	Negativo (Normal)	**FP**	**TN**

A partir dos valores presentes na matriz de confusão, algumas métricas são derivadas:

- Taxa de verdadeiros positivos (TPR), ou sensibilidade, que considera todos os casos positivos corretamente classificados sobre o total de casos positivos.

$$TPR = \frac{TP}{TP + FN} \tag{5.1}$$

- Taxa de falsos positivos (FPR), que considera todos os falsos positivos sobre o total de casos negativos.

$$FPR = \frac{FP}{TN + FP} \quad (5.2)$$

- Precisão, que considera os casos positivos corretamente classificados sobre o total de casos classificados como positivos.

$$Precisão = \frac{TP}{TP + FP} \quad (5.3)$$

- Medida-F (F-score), é uma forma de medir a acurácia do sistema que leva em consideração a precisão e a sensibilidade. É uma métrica importante pois permite calcular o equilíbrio entre essas duas medidas.

$$F - score = 2 * \frac{Precisão * TPR}{Precisão + TPR} \quad (5.4)$$

Além das métricas (5.1), (5.2), (5.3) e (5.4), em alguns casos são utilizadas curvas ROC. A curva ROC é uma abordagem estatística padrão para avaliar detectores ou classificadores binários. A curva ROC possibilita a visualização do balanço entre a TPR (5.1) e a FPR (5.2) em uma única curva, em função de um limiar.

5.3 Experimentos utilizando o conjunto de dados NSL-KDD

Nesta seção são relatados os resultados obtidos a partir dos experimentos realizados utilizando o conjunto de dados NSL-KDD (TAVALLAEE et al., 2009). O conjunto de dados NSL-KDD é uma versão aprimorada do conjunto de dados KDD'99. O conjunto KDD'99 é derivado do programa *DARPA Intrusion Detection Evaluation* de 1998 realizado pelo *MIT Lincoln Labs* e tem sido utilizado por pesquisadores que estudam o uso de algoritmos de aprendizagem automática na detecção de intrusos. Embora amplamente utilizado, o conjunto de dados KDD'99 apresenta uma série de

deficiências (MCHUGH, 2000). O conjunto NSL-KDD resolveu alguns problemas do conjunto original, como dados redundantes.

Os dados que compõem base NSL-KDD são divididos em conjuntos de treinamento e conjuntos de teste. Uma vez que o método proposto não necessita passar por uma fase de treinamento *off-line*, seu desempenho foi avaliado utilizando a base de treinamento que contém uma quantidade significativamente maior de conexões. O conjunto de dados utilizado, KDDTrain+, possui 125.973 conexões, das quais 67.343 (53,4%) são conexões normais e 58.630 são ataques. Os ataques são divididos em 4 categorias, sendo elas: ataques de reconhecimento (probe), ataques de negação de serviço (DoS), acesso não autorizado a partir de um host remoto (R2L) e acesso não autorizado a funções de administrador (U2R).

Cada conexão é representada por 41 características, sendo elas 34 contínuas e as outras 7 simbólicas. Neste trabalho foram utilizadas apenas as características contínuas, pois as métricas utilizadas para o cálculo das distâncias não suportam atributos simbólicos.

5.3.1 Pré-processamento dos dados

Seguindo a premissa de que anomalias são eventos raros, foram utilizadas 67.343 conexões normais e apenas 8.000 ataques, totalizando 75.343 registros analisados. As amostras de ataque foram escolhidas aleatoriamente, mantendo a mesma proporção de cada classe de ataque em relação ao conjunto original.

Após realizar a amostragem dos dados, os atributos das conexões foram normalizados. A normalização é um processo importante para que nenhum atributo se sobressaia em relação aos demais. O objetivo é alcançar o mesmo poder de influência para atributos em escalas diferentes, de modo que seja possível compará-los. Por exemplo, suponha que a duração de uma conexão possa variar entre 0ms e 10.000ms enquanto que a taxa de conexões sem resposta pode possuir um valor entre 0 e 1. Ao realizar o cálculo da distância entre o objeto e o centro do grupo sem uma normalização destes

valores, frequentemente o atributo que diz respeito à duração irá se sobressair. Os dados foram normalizados utilizando a equação (5.5)

$$x_i^k = \frac{v_i^k - \mu^k}{\sigma^k} \tag{5.5}$$

onde μ^k é o valor médio do atributo k e σ^k é seu desvio padrão. A partir desta normalização, os atributos passam a ter média zero e desvio padrão 1.

Dentre os processos de normalização existentes, a normalização escolhida apresenta melhores resultados para a utilização em algoritmos de agrupamento (WANG et al., 2014), em especial o algoritmo k-médias (MOHAMAD; USMAN, 2013).

5.3.2 Avaliação do algoritmo DBSCAN

O sucesso do algoritmo de detecção proposto depende fortemente do desempenho do algoritmo DBSCAN. Dessa forma, foi realizada uma análise do comportamento deste algoritmo em relação ao conjunto de dados utilizado. Dentre os parâmetros utilizados pelo DBSCAN, dois deles estão diretamente relacionados a sua definição de grupos: o número mínimo de objetos que um grupo deve ter ($minPts$) e a distância máxima entre dois objetos para que estes sejam considerados vizinhos (eps). Para a realização dos testes utilizando o algoritmo DBSCAN. O conjunto de dados foi processado em intervalos com 500 registros cada. O parâmetro $minPts$ na primeira execução foi igual a 4, sendo dobrado a cada execução até o valor de 256. Já o parâmetro eps inicialmente teve seu valor igual a 1, sendo incrementado em 0,5 a cada execução até valor de 3. A taxa de detecção (TPR) e a taxa de falsos positivos (FPR) para cada combinação destes valores são apresentadas na Tabela 5.2.

Tabela 5.2 Análise da NSL-KDD utilizando apenas o DBSCAN.

eps	1		1,5		2		2,5		3	
minPts	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
4	83,8%	34,8%	87,1%	22,6%	81,0%	18,3%	76,2%	16,1%	78,1%	13,1%
8	99,6%	40,1%	98,4%	24,1%	97,2%	19,1%	96,2%	16,8%	94,8%	14,4%
16	99,7%	48,7%	98,7%	29,6%	97,9%	21,1%	96,6%	17,3%	95,2%	15,3%
32	100%	98,5%	99,2%	42,0%	98,4%	25,0%	97,1%	19,1%	96,0%	15,8%
64	100%	100%	99,4%	56,6%	98,6%	31,0%	97,8%	22,5%	96,1%	17,1%
128	100%	100%	100%	97,8%	98,8%	35,6%	98,1%	24,9%	96,4%	18,1%
256	100%	100%	100%	100%	100%	100%	98,8%	48,8%	97,9%	23,7%

É possível notar que a TPR e a FPR aumentam quanto maior o valor de $minPts$. Isso ocorre porque grupos pequenos, que possuem quantidade de elementos menor que $mintPts$, acabam sendo identificados como ruídos e, consequentemente, como anomalias, causando uma taxa elevada de falsos positivos. Em compensação, quando o valor de eps é maior, tanto a TPR quanto a FPR são menores. Com um valor de eps maior, mais elementos são considerados próximos e acabam sendo agrupados, abrangendo um número maior de elementos em um mesmo grupo.

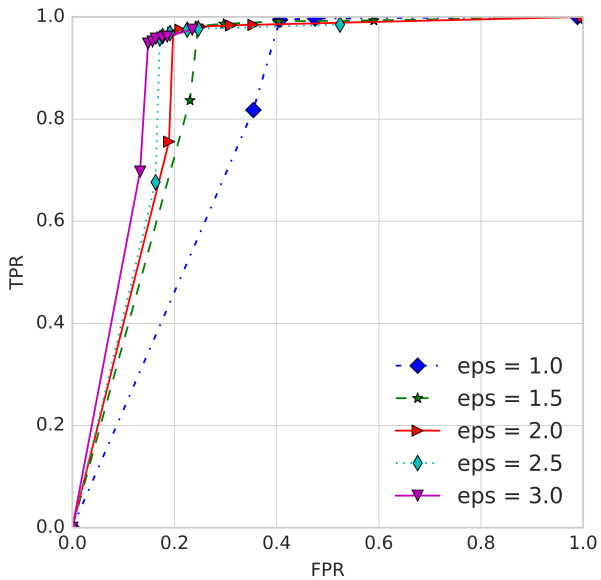


Figura 5.1 Análise do conjunto NSL-KDD utilizando o algoritmo DBSCAN com diferentes valores de eps.

Na Figura 5.1 são ilustradas as curvas ROC referente às execuções do algoritmo DBSCAN com diferentes valores para o parâmetro eps. É possível observar que o algoritmo possui curvas similares para valores entre 2 e 3. Para valores acima de 3 há um aumento significativo na taxa de falsos negativos. Já na Figura 5.2 é ilustrado o comportamento do algoritmo DBSCAN com diferentes valores para $minPts$. Para valores de $minPts$ acima de 8, o algoritmo apresentou comportamento semelhante nas diferentes execuções, apresentando uma degradação da relação entre TPR e FPR com $minPts$ igual a 256. O motivo desta semelhança é que os grupos formados por anomalias possuem uma quantidade pequena de elementos, corroborando com a premissa de que anomalias são eventos raros presentes no conjunto de dados. As conexões anômalas somente formarão um grupo na ocorrência de ataques volumosos, como negativas de serviço por inundação.

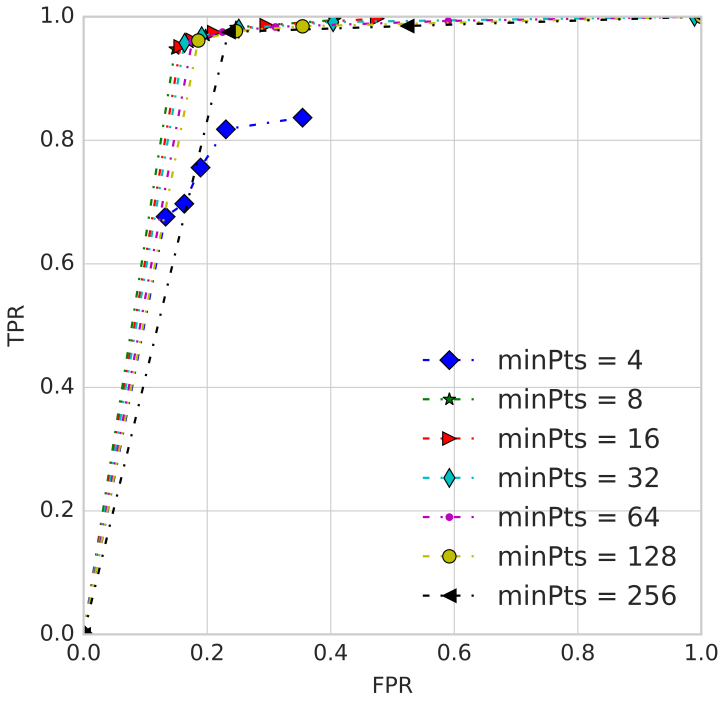


Figura 5.2 Análise do conjunto NSL-KDD utilizando o algoritmo DBSCAN com diferentes valores de $minPts$.

Outro parâmetro do sistema que influencia no algoritmo DBSCAN é o parâmetro δ, utilizado como limiar para definir se um grupo encontrado é um grupo de conexões normais ou de conexões anômalas, conforme explicado na seção 4.5. A RDc ideal para cada grupo é 1, isto é, todos os grupos têm a mesma quantidade de objetos. Observando os testes realizados, os grupos normais encontrados pelo DBSCAN possuíam RDc acima de 0,8, enquanto que os grupos anômalos possuíam RDc abaixo de 0,3. Considerando possíveis variações, o valor de δ utilizado no algoritmo de detecção foi de 0,5, isto é, um grupo deve ter pelo menos a metade de elementos do que o esperado.

5.3.3 Parâmetros do algoritmo de detecção

O método proposto possui alguns parâmetros que podem ser modificados para refinar seu desempenho. Nesta seção são realizadas algumas considerações em relação aos valores destes parâmetros e seu impacto na detecção de anomalias. Os parâmetros $minPts$, eps e δ foram abordados na seção 5.3.2. Para a realização dos experimentos a seguir, foram utilizados $minPts$ igual a 8, eps igual a 1,5 e δ igual a 0,5.

O parâmetro k define o número de grupos utilizado pelo algoritmo k-médias, o valor ideal para k é a quantidade de classes existentes dentro do tráfego normal. Entretanto, definir a quantidade de classes dentro do conjunto normal é uma tarefa difícil e tem sido explorada em outras pesquisas (ERMAN; ARLITT; MAHANTI, 2006; NGUYEN; ARMITAGE, 2008; ZANDER; NGUYEN; ARMITAGE, 2005). Neste trabalho é utilizado o método apresentado em (PHAM; DIMOV; NGUYEN, 2005) para a determinação do valor de k, sendo $k = 4$. Embora o valor de k seja difícil de ser estabelecido, um experimento realizado mostra que o método proposto não depende fortemente do valor de k. Na Figura 5.3 são apresentados os resultados obtidos com a execução do método proposto utilizando diversos valores para o número de grupos k.

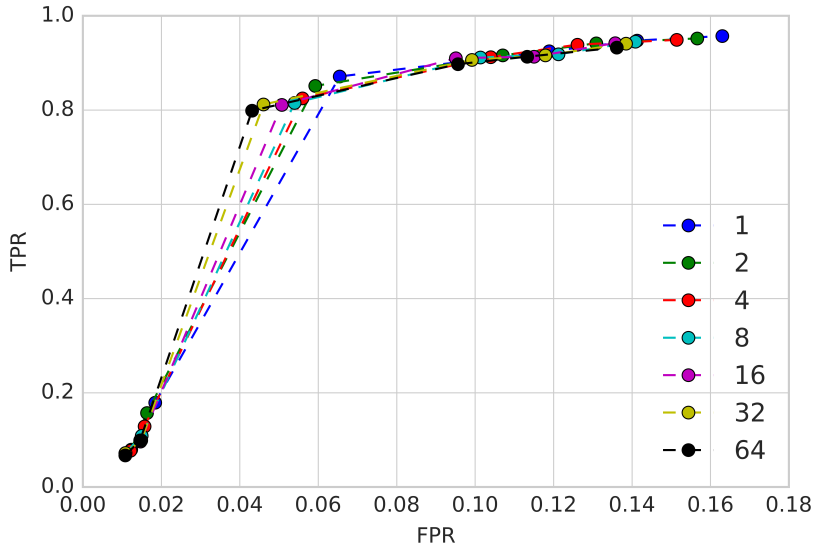


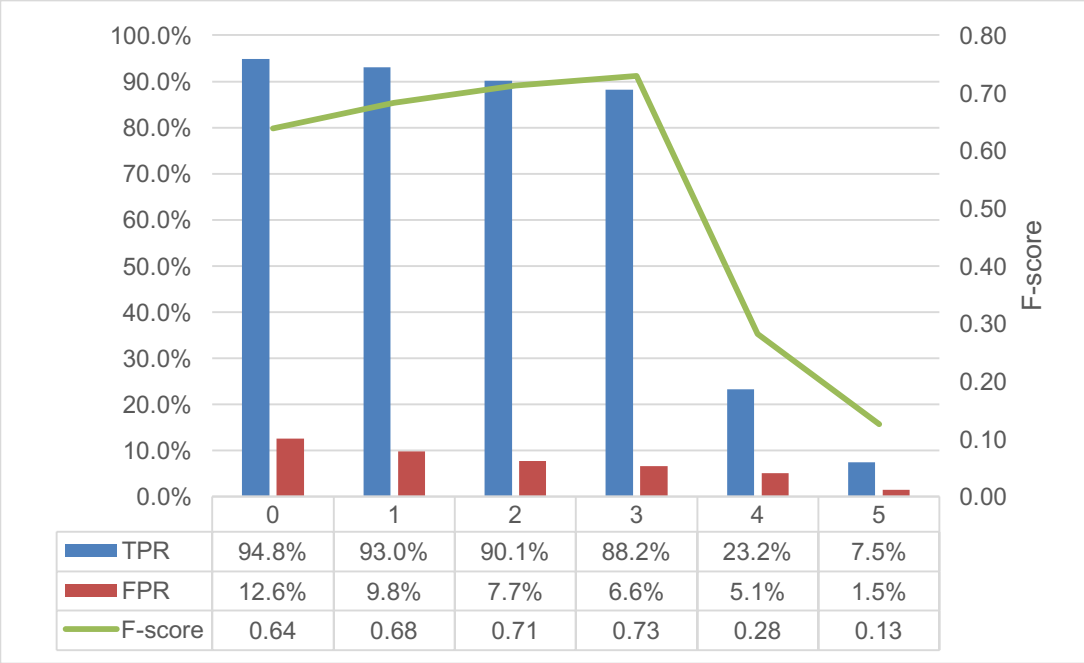
Figura 5.3 NSL-KDD – Utilizando diferentes valores de k no algoritmo k-médias.

É possível observar que há uma variação sutil nos resultados entre cada valor de k. Isto ocorre porque apenas os dados considerados normais durante a primeira etapa são utilizados. Independentemente do número de grupos k especificado, os centroides são calculados de acordo com os grupos normais encontrados pelo algoritmo DBSCAN. Dessa forma, caso seja especificado um número maior de grupos do que o necessário para representar as classes normais, o que irá ocorrer é apenas a divisão em um grupo normal em grupos menores. Outra abordagem possível para a definição do valor de k é a observação do número de grupos encontrados pelo algoritmo DBSCAN no decorrer de algumas análises.

Os parâmetros n e α estão relacionados à distribuição dos dados analisados. O valor de n depende da velocidade do fluxo analisado e da capacidade de processamento. Já o valor de α depende da velocidade de mudança no comportamento da rede, quanto mais rápida a alteração no comportamento normal, menor deve ser o valor de α. Para a execução dos testes utilizando o conjunto de dados NSL-KDD utilizou-se n igual a 500 e α igual a 1.

Por fim, os parâmetros τ_1 e τ_2 são utilizados apenas quando a distância TEV, abordada na seção 4.7, é escolhida. O parâmetro τ_1 define o valor

mínimo que cada atributo de x deve diferir de seu exemplar para que este seja somado no TEV.

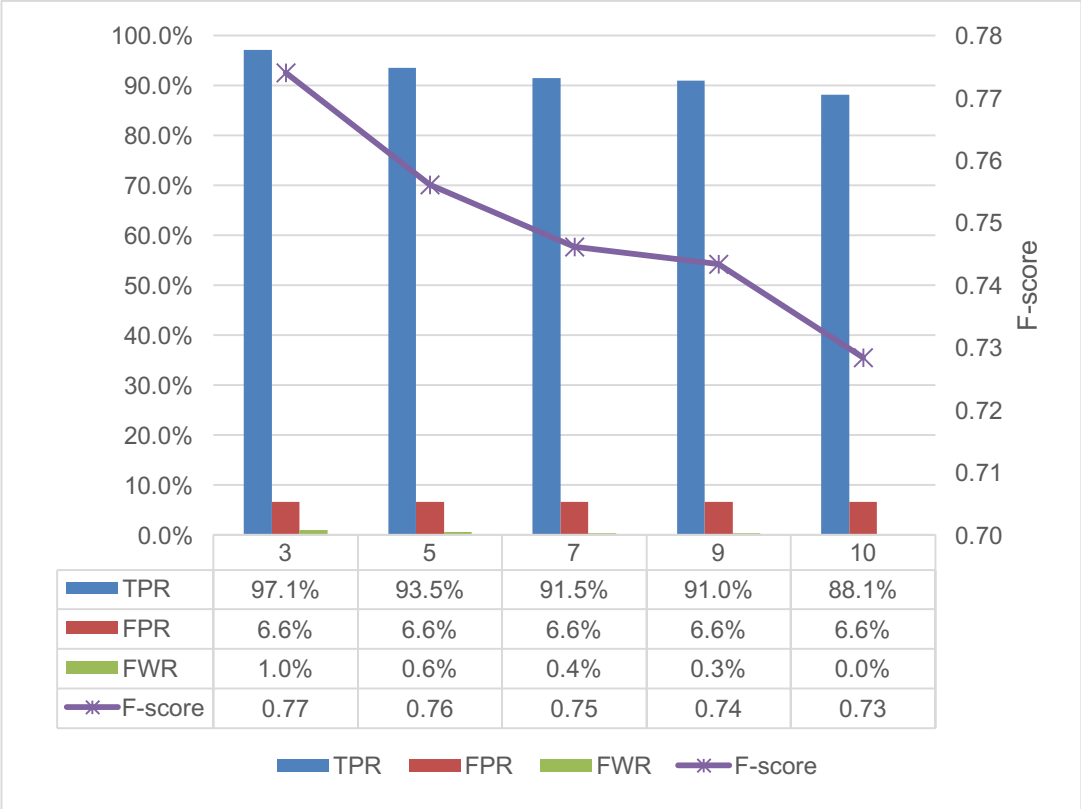


	0	1	2	3	4	5
TPR	94.8%	93.0%	90.1%	88.2%	23.2%	7.5%
FPR	12.6%	9.8%	7.7%	6.6%	5.1%	1.5%
F-score	0.64	0.68	0.71	0.73	0.28	0.13

Figura 5.4 NSL-KDD - Resultados obtidos para diferentes valores de τ_1.

Na Figura 5.4 são apresentados os resultados das execuções com diferentes valores de τ_1, considerando o limiar ε igual 10 – obtido empiricamente através de experimentos –, sem o mecanismo de encaminhamento. Nota-se que para valores de τ_1 acima de 3, há uma degradação significativa no desempenho de detecção. Isso porque desvios significativos para a detecção das anomalias são desconsiderados.

Já o parâmetro τ_2 estabelece o limiar inferior para que os elementos sejam considerados suspeitos, influenciando na quantidade de elementos encaminhados para análise externa. Considerando os valores de τ_1 e ϵ iguais a 3 e 10, respectivamente, foram realizados testes com diferentes valores para τ_2. Como pode ser observado na Figura 5.5, quanto mais próximo o valor de τ_2 do limiar ε, menor a taxa de detecção (TPR). Isso ocorre porque mais elementos anômalos são considerados normais, aumentando a taxa de falsos negativos. O ideal é encontrar um balanço entre a taxa de detecção e a quantidade de elementos encaminhados.



	3	5	7	9	10
TPR	97.1%	93.5%	91.5%	91.0%	88.1%
FPR	6.6%	6.6%	6.6%	6.6%	6.6%
FWR	1.0%	0.6%	0.4%	0.3%	0.0%
F-score	0.77	0.76	0.75	0.74	0.73

Figura 5.5 NSL-KDD - Desempenho com diferentes valores de τ_2.

Na Figura 5.5 é apresentada, além dos valores de TPR, FPR e F-score, a taxa de objetos encaminhados para análise externa (FWR). O caso em que τ_2 é igual a 3, isto é $\tau_2 = \tau_1$, apresenta o maior valor para o índice F-score. Com aproximadamente 1% de todos os registros encaminhados, este caso apresenta um aumento de 10,2% na taxa de detecção em relação ao caso onde $\tau_2 = \varepsilon$.

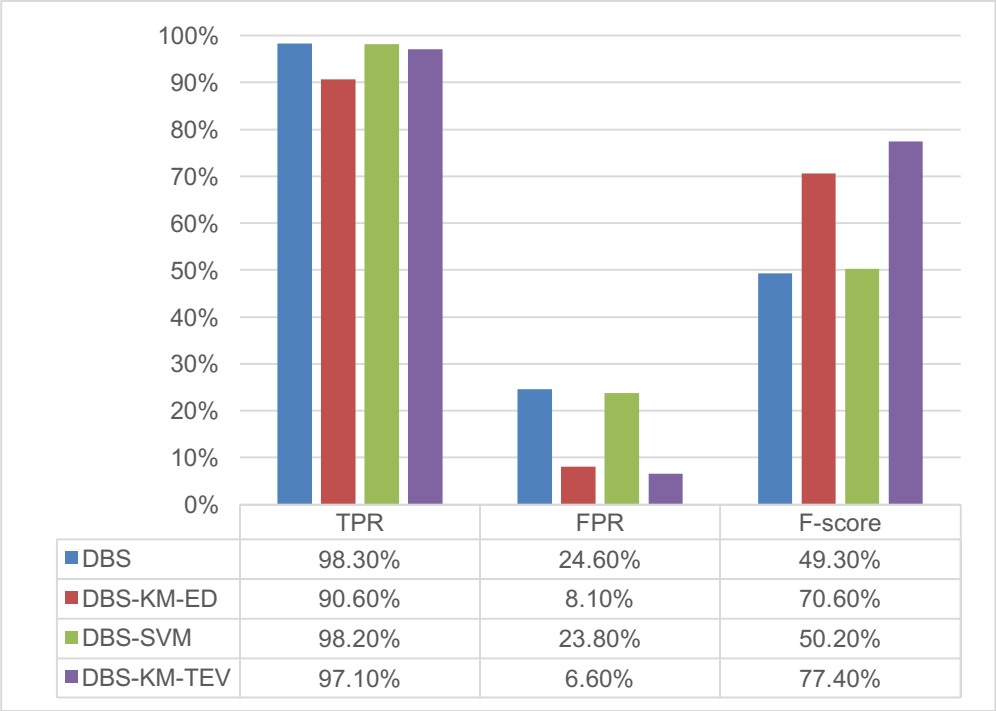
5.3.4 Comparação dos resultados obtidos

A fim de avaliar o desempenho de detecção do método proposto, foram comparadas quatro abordagens. A primeira abordagem, chamada de DBS, é a utilização apenas do algoritmo DBSCAN, processando 500 objetos a cada iteração. A segunda abordagem, chamada de DBS-KM-ED, é a combinação do algoritmo DBSCAN com o algoritmo k-médias, utilizando-se a distância euclidiana como critério de decisão na detecção. Por fim, a abordagem DBS-

KM-TEV é a combinação do algoritmo DBSCAN com o algoritmo k-médias utilizando-se o valor de desvio do limiar como critério de decisão e a utilização do mecanismo de encaminhamento. Ainda, para fins de comparação, é apresentado o desempenho da combinação entre o algoritmo DBSCAN e o algoritmo *one-class* SVM, chamado de DBS-SVM.

O algoritmo *one-class* SVM, apresentado por Schölkopf et al. (SCHÖLKOPF et al., 2001), é um algoritmo não supervisionado também utilizado para a detecção de anomalias. Dado um conjunto de dados, o algoritmo utiliza uma função $\phi(x)$ para mapear os dados em um espaço H, de modo que o produto escalar possa ser calculado utilizando um núcleo (*kernel*) $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. Após o mapeamento, encontra-se um hiperplano capaz de separar os dados com a maior margem possível, chegando a um problema de otimização. A função de decisão $f(x)$ é alcançada resolvendo um problema de otimização, onde o retorno desta função é 1, para dados contidos dentro da fronteira, e -1 para todo o resto. Na detecção de anomalias, um conjunto de dados normais é utilizado para se estabelecer a função $f(x)$. Dado um objeto do conjunto de testes, se o resultado for positivo, ele é considerado normal. Caso contrário, ele é considerado uma anomalia.

Nos testes realizados utilizando o algoritmo *one-class* SVM, os pontos considerados normais após a primeira etapa – agrupamento utilizando o DBSCAN – são utilizados para o treinamento do algoritmo, isto é, para encontrar a função $f(x)$. Os dados considerados anômalos são então classificados de acordo com o modelo obtido.



	TPR	FPR	F-score
DBS	98.30%	24.60%	49.30%
DBS-KM-ED	90.60%	8.10%	70.60%
DBS-SVM	98.20%	23.80%	50.20%
DBS-KM-TEV	97.10%	6.60%	77.40%

Figura 5.6 NSL-KDD - Comparação entre diferentes abordagens.

Na Figura 5.6 é apresentado o desempenho de cada uma das abordagens apresentadas neste trabalho. A combinação dos algoritmos DBSCAN e *one-class* SVM não apresentou melhora significativa em relação a utilização do algoritmo DBSCAN isoladamente. O motivo é que o processo de treinamento do algoritmo *one-class* SVM consiste em definir uma fronteira que separe os dados normais das anomalias e, dados que o treinamento foi realizado com os elementos considerados normais pelo DBSCAN, o algoritmo não foi capaz de detectar os falsos positivos apresentados pelo DBSCAN. Já a utilização do algoritmo k-médias produziu resultados significativos na redução dos falsos positivos. A utilização da métrica TEV em conjunto com o mecanismo de encaminhamento apresentou uma taxa de detecção próxima ao algoritmo DBSCAN, com uma redução de aproximadamente 73% na taxa de falsos positivos, com apenas 1% das conexões encaminhadas para análise externa.

5.4 Conjunto de dados ACME-IDE

O uso de conjuntos de dados muito datados, tais como o DARPA 1998 e KDD'99, deve ser evitado a fim de se obter resultados consistentes em relação às atividades de rede observadas atualmente. Tais conjuntos sofrem pela falta de atualização perante suas atividades de rede, tanto em relação aos tipos de atividades maliciosas, quanto ao tráfego legítimo de rede, isso quando disponível. Ainda, apresentam uma distribuição não realista dos ataques, utilizando a mesma proporção entre tráfego normal e ataques, e executam os ataques isoladamente e depois misturam com tráfego legítimo, de modo que é visível a diferença de comportamento entre os dados. Assim, tais conjuntos não podem ser considerados unicamente no desenvolvimento e avaliação de sistemas de segurança.

Embora o conjunto NSL-KDD tenha aprimorado o conjunto KDD'99, outros problemas persistem, como a falta de representação de ataques mais modernos e cenários reais de ataque. Nenhuma validação é realizada para garantir que o conjunto de dados da KDD'99 é similar a um tráfego de rede real. Para não tirar conclusões sobre a metodologia apresentada neste trabalho utilizando apenas o conjunto de dados NSL-KDD, foram realizados testes utilizando o conjunto de dados ACME-IDE (ACME *Intrusion Detection Evaluation*) desenvolvido no laboratório ACME!, onde esta pesquisa foi realizada.

O conjunto de dados ACME-IDE, trata-se de um conjunto mais atual que busca uma representação fidedigna do tráfego de rede. A mesma conta com diversas características, essenciais para o desenvolvimento do trabalho em questão. Dentre os objetivos da criação deste conjunto, o principal é prover cenários de ataque com ambientes de redes atualizados, tanto em relação às atividades maliciosas de rede, quanto em atividades legítimas de usuários.

O conjunto de dados em questão se destaca por conter atividades de redes geradas especificamente para o propósito de avaliação de sistemas de segurança. Dessa forma, não há a necessidade de sanitização de seus dados, o que mantem inalterado diversas informações, que podem ser críticas dependendo dos sistemas que utilizarão estes dados. Ainda, as atividades de

rede maliciosas foram geradas de forma simultânea à geração de tráfego normal, proporcionando um cenário realista de rede. Com essa abordagem, não é necessária nenhuma alteração na estrutura dos dados para se obter outros tipos de comportamento, tal como realizado em outros conjuntos, evitando assim, problemas de integridade que possam ser gerados.

5.4.1 Atividades contidas no conjunto

O conjunto foi desenvolvido no ano de 2016 e conta com diversas ferramentas atualizadas em sua geração. Essas ferramentas foram selecionadas a fim de obter a reprodutibilidade do conjunto de dados, atendendo a requisitos como estabilidade e previsibilidade. Dessa forma, foi necessário analisar o comportamento específico de rede de cada ferramenta em diversas execuções, obtendo seu comportamento específico de rede. Na Tabela 5.3 há uma sumarização dos tipos de atividades maliciosas inseridas no conjunto no decorrer de seu desenvolvimento.

Tabela 5.3 Atividades maliciosas executadas para geração do conjunto ACME-IDE.

Classe de ataque	*Variações do tipo de ataque*	*Quantidade de execuções*
Scan	22	97
Vulnerability Scan	8	15
DOS	5	6
Bruteforce	8	22
Spoofing	1	2
TOTAL	**44**	**142**

5.4.2 Geração do conjunto de dados

As atividades de redes foram geradas em um ambiente virtualizado, contendo diversos equipamentos que provêm serviços de rede, a fim de reproduzir um ambiente real. Dentre as atividades simuladas dentro do

ambiente estão: tráfego HTTP/HTTPS, requisições DNS, transferência de arquivos via FTP, SSH, SMB, e HTTP, além de conexões SSH e de acesso remoto via VNC. Todas as atividades foram geradas por ferramentas atualizadas e sistemas operacionais atuais. Entre os sistemas utilizados há uma variedade de versões do Windows, a partir do Windows XP até o Windows 10, e sistemas Linux com versão de kernel entre 2.6 e 3.16, a saber, sistemas Ubuntu Linux da versão 10.04 até a versão 14.04 e Kali Linux 2.

As atividades legítimas de rede foram geradas de forma constante, enquanto que, as atividades maliciosas foram agendadas e catalogadas para a sua futura classificação. Durante o período de 5 dias de geração dos dados, foram obtidos um total de 10,2 GB de captura completa do tráfego de rede. Os arquivos de captura foram divididos por dia, onde cada dia conta com uma documentação específica, contendo informações sobre cada ataque realizado. A Tabela 5.4 contém uma sumarização dos dados gerados diariamente.

Tabela 5.4 ACME-IDE - Distribuição dos dados gerados por dia.

Dia de Teste	Quantidade de Ataques	Total de Conexões	Tamanho da Captura
16/05/2016	44	52.105	1.5 GB
17/05/2016	24	93.443	2.1 GB
18/05/2016	16	77.802	1.1 GB
19/05/2016	23	137.621	3.4 GB
20/05/2016	35	89.055	2.1 GB
Total	**142**	**450.026**	**10.2 GB**

5.4.3 Validação dos dados obtidos

A validação e análise dos resultados gerados são imprescindíveis para determinar a qualidade do conjunto de dados. As atividades executadas foram validadas baseou-se na utilização de ferramentas de análise de pacotes, tal

como *WIRESHARK*, com o qual foi possível observar a ocorrência dos ataques nos devidos horários estipulados.

Outra maneira de se validar a base quanto a ocorrência correta dos ataques se deu através de testes práticos utilizando IDSs baseados em assinaturas, tal como *SNORT*. A partir de assinaturas bem definidas sobre os tipos de ataques abordados, com a análise dos alertas gerados, é possível verificar de fato a ocorrência das atividades descritas.

A verificação das atividades não maliciosas, uma vez que não são agendadas em horários específicos, somente pode ser feita através da análise estatística dos dados gerados. Uma característica que pode ser observada a partir da Tabela 5.5 é a quantidade predominante de tráfego TCP em relação aos outros protocolos da camada de transporte. Essa característica se mostra presente devido ao tipo de tráfego simulado.

Tabela 5.5 ACME-IDE - Distribuição do tráfego de rede de acordo com o protocolo de transporte.

Protocolo	**Distribuição**
TCP	92.32%
UDP	5.28%
Outros	1.98 %

Outra característica importante de se observar é a distribuição do tráfego entre os protocolos de camada de aplicação (Tabela 5.6), sendo o tráfego WEB o de maior quantidade, seguido de tráfegos como SSH e DNS. Essa distribuição condiz com o comportamento típico observado em um ambiente de rede corporativo, protegido por um *firewall*, e assim podemos constatar a eficiência da base quanto à presença de tráfego de fundo.

Tabela 5.6 ACME-IDE - Distribuição do tráfego de rede de acordo com o protocolo de aplicação.

Protocolo	Distribuição
80 - HTTP	61.04 %
443 - HTTPS	21.52 %
22 - SSH	7.15 %
53 - DNS	4.58 %
445 - SMB	2.90 %
Outros	2.81 %

5.5 Experimentos realizados utilizando o conjunto ACME-IDE

Os testes foram conduzidos de maneira similar aos testes realizados com o conjunto de dados NSL-KDD. Primeiramente, os dados foram processados a fim de extrair características relevantes para a distinção do tráfego normal e dos ataques. As características extraídas foram então normalizadas e processadas utilizando o algoritmo DBSCAN para a avaliação dos parâmetros. Após a definição dos parâmetros do algoritmo DBSCAN, foram realizados os testes utilizando o método proposto, utilizando tanto a distância euclidiana como a distância TEV. Os resultados de cada método são então comparados a fim de validar o desempenho do algoritmo proposto.

5.5.1 Pré-processamento dos dados

O conjunto de dados ACME-IDE contém informações sobre cada conexão realizada dentro do período de testes. Para a realização dos experimentos a seguir, foram utilizados os dados coletados no primeiro dia do conjunto, composto por 52.105 conexões, das quais 9.538 são conexões geradas pelos 44 ataques executados durante o dia de coleta.

A fim de extrair algumas características de comportamento, foi realizado um resumo das informações das conexões considerando endereços IP de origem distintos em intervalos de 30 segundos. A partir deste agrupamento foram extraídas as características utilizadas para detecção, conforme explicado na seção 4.4.

O agrupamento realizado reduziu consideravelmente a quantidade de dados a serem processados, conforme pode ser observado na Tabela 5.7. Os dados normais foram reduzidos em uma taxa de 4 para 1, enquanto que os dados de ataques foram reduzidos em uma taxa de 136 para 1. Essa grande redução ocorre pelo fato de que algumas atividades, como varreduras de rede e serviços, geram uma grande quantidade de conexões e são agrupadas em uma única linha.

Tabela 5.7 Redução dos dados utilizando o agrupamento descrito.

	Antes do agrupamento	*Após o agrupamento*
Dados normais	42.567	10.132
Dados de ataques	9.538	70
Total	52.105	10.202

Após a extração das características, os dados foram normalizados utilizando a equação (5.5) apresentada na seção 5.3.1.

5.5.2 Avaliação do algoritmo DBSCAN com o conjunto ACME-IDE

Após a extração e normalização das características do conjunto ACME-IDE, os dados foram agrupados utilizando o algoritmo DBSCAN com diferentes valore para os parâmetros $minPts$ e eps. O parâmetro $minPts$ foi utilizado com valor inicial igual 2, dobrando a cada execução até o valor de 128. Já o parâmetro eps teve seu valor inicial igual a 0,01, sendo incrementado em 0,4 após a primeira execução e 0,5 nas demais. O parâmetro δ, utilizado para definir se um grupo possui uma densidade abaixo do esperado, teve seu

valor igual a 0,5 em todas as execuções. Já o tamanho da janela de dados processados foi definido igual a 1000. Na Tabela 5.8 são apresentadas a TPR e a FPR de cada combinação dos valores utilizados.

Tabela 5.8 Análise do conjunto ACME-IDE utilizando apenas o DBSCAN.

eps	**0,01**		**0,05**		**0,1**		**0,15**		**0,2**	
minPts	**TPR**	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**
2	100%	39,1%	97,1%	26,3%	94,3%	13,3%	91,4%	12,9%	87,1%	12,6%
4	100%	46,4%	97,1%	32,2%	94,3%	22,1%	91,4%	18,8%	87,1%	16,0%
8	100%	60,0%	97,1%	49,5%	94,3%	27,6%	92,9%	21,6%	87,1%	18,3%
16	100%	72,3%	97,1%	56,6%	94,3%	33,0%	92,9%	24,4%	88,6%	21,1%
32	100%	84,0%	98,6%	58,4%	97,1%	45,8%	95,7%	33,3%	92,9%	32,4%
64	100%	95,8%	100%	100%	97,1%	55,6%	95,7%	38,6%	94,3%	33,0%
128	100%	100%	100%	100%	100%	79,2%	95,7%	47,8%	94,3%	36,7%

Observando a Tabela 5.8, observa-se a mesma relação entre os parâmetros eps e $minPts$ presente na análise do conjunto de dados NSL-KDD. Na Figura 5.7 é possível notar que o algoritmo DBSCAN teve seu melhor desempenho com o parâmetro eps igual a 0,1.

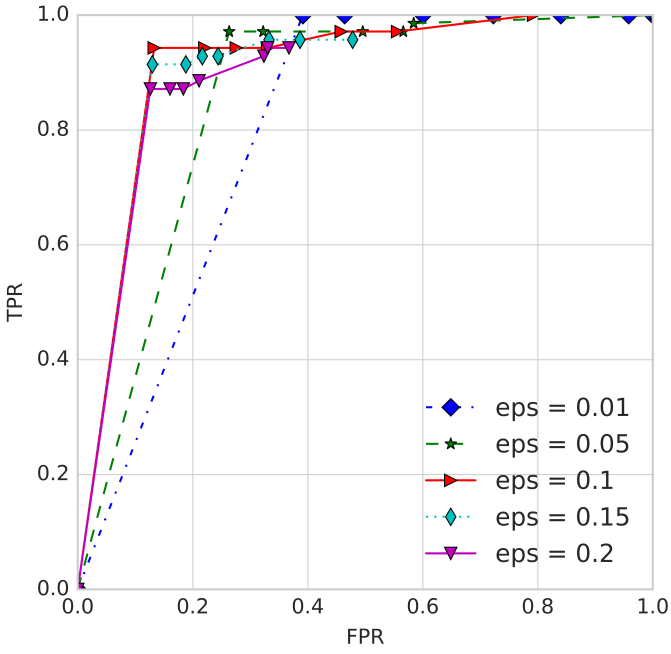


Figura 5.7 Análise do conjunto ACME-IDE utilizando o algoritmo DBSCAN com diferentes valores de eps.

Em relação ao parâmetro $minPts$, o melhor desempenho, conforme ilustrado na Figura 5.8, foi obtido com um valor igual a 2, quatro vezes menor do que o utilizado durante a análise do conjunto NSL-KDD. Essa diferença ocorre principalmente pelo agrupamento realizado durante a extração do conjunto de características.

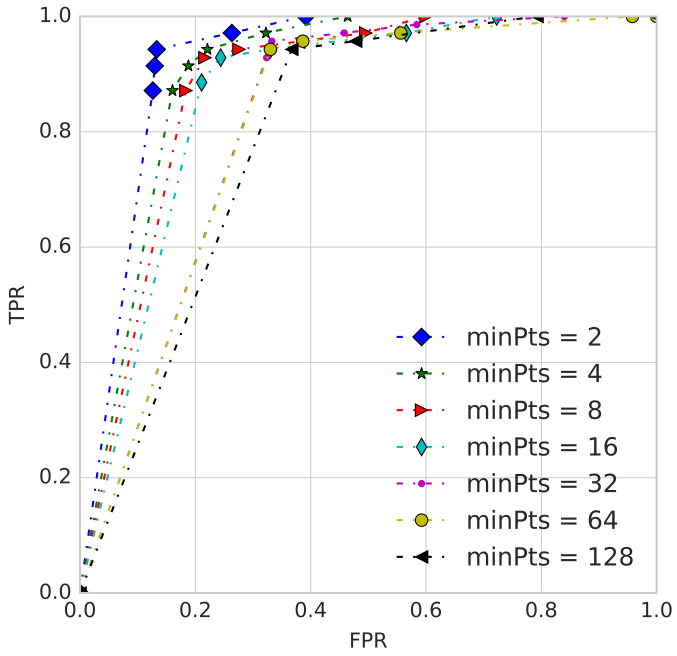


Figura 5.8 Análise do conjunto ACME-IDE utilizando o algoritmo DBSCAN com diferentes valores de $minPts$.

5.5.3 Análise da metodologia proposta

Nesta seção são apresentados os resultados obtidos após a execução da metodologia proposta utilizando diferentes valores para os parâmetros. Os parâmetros $minPts$, eps e δ receberam os valores 2, 0,1 e 0,5, respectivamente, conforme descrito na seção 0. Para a execução dos testes utilizando o conjunto de dados ACME-IDE utilizou-se n igual a 1000 e α igual a 1.

O parâmetro k foi definido a partir da análise dos grupos gerados pelo algoritmo DBSCAN. No total, foram encontrados 14 grupos considerados

normais. Entretanto, conforme discutido na seção 5.3.3, a metodologia proposta é capaz de lidar com diferentes valores de k de maneira semelhante.

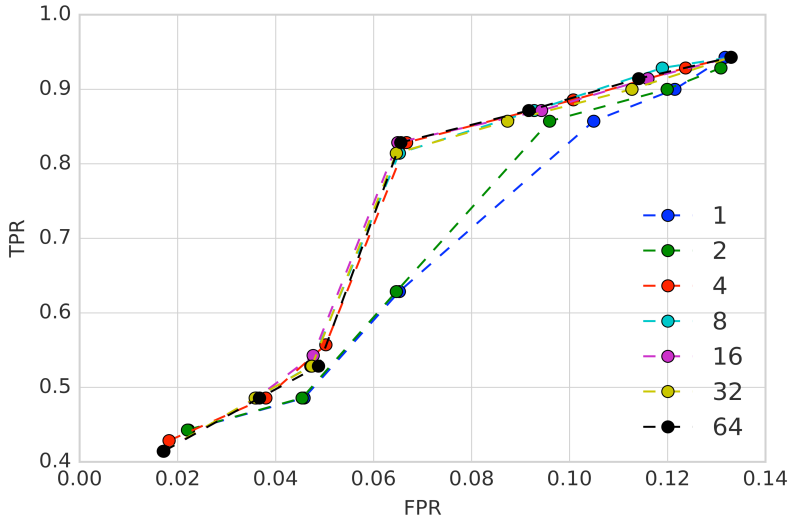
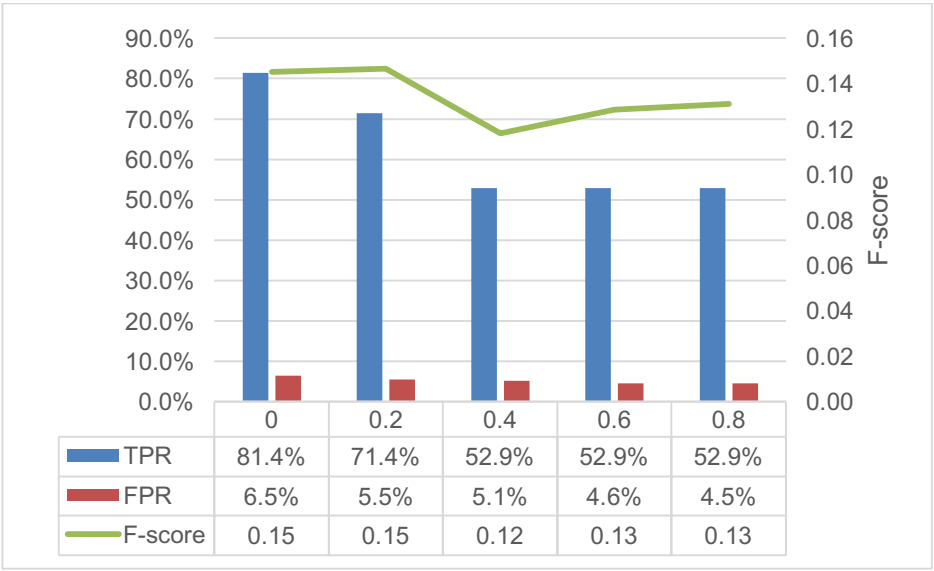


Figura 5.9 ACME-IDE - Algoritmo k-médias utilizando diferentes valores de k.

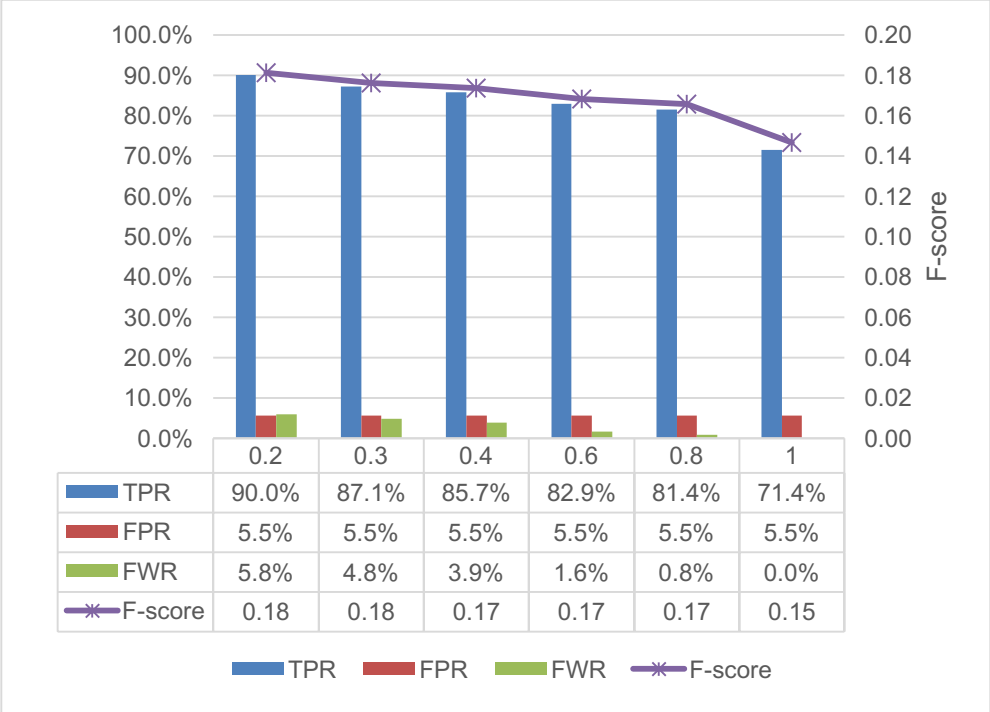
Na Figura 5.9 são apresentadas as curvas ROC das diferentes execuções variando o valor do parâmetro k. Como pode ser observado, é necessário um número mínimo de grupos para representar o tráfego normal. Por fim, foram realizados experimentos com diferentes valores para os parâmetros τ_1 e τ_2, considerando o limiar ε igual a 1.



	0	0.2	0.4	0.6	0.8
TPR	81.4%	71.4%	52.9%	52.9%	52.9%
FPR	6.5%	5.5%	5.1%	4.6%	4.5%
F-score	0.15	0.15	0.12	0.13	0.13

Figura 5.10 ACME-IDE - Resultados obtidos para diferentes valores de τ_1.

Na Figura 5.10 são apresentados os resultados das execuções com diferentes valores de τ_1, considerando o limiar ε igual 1, sem a utilização do mecanismo de encaminhamento. O melhor balanço entre TPR e FPR, isto é, o maior valor para o índice F-score é obtido com τ_1 igual a 0,2.



	0.2	0.3	0.4	0.6	0.8	1
TPR	90.0%	87.1%	85.7%	82.9%	81.4%	71.4%
FPR	5.5%	5.5%	5.5%	5.5%	5.5%	5.5%
FWR	5.8%	4.8%	3.9%	1.6%	0.8%	0.0%
F-score	0.18	0.18	0.17	0.17	0.17	0.15

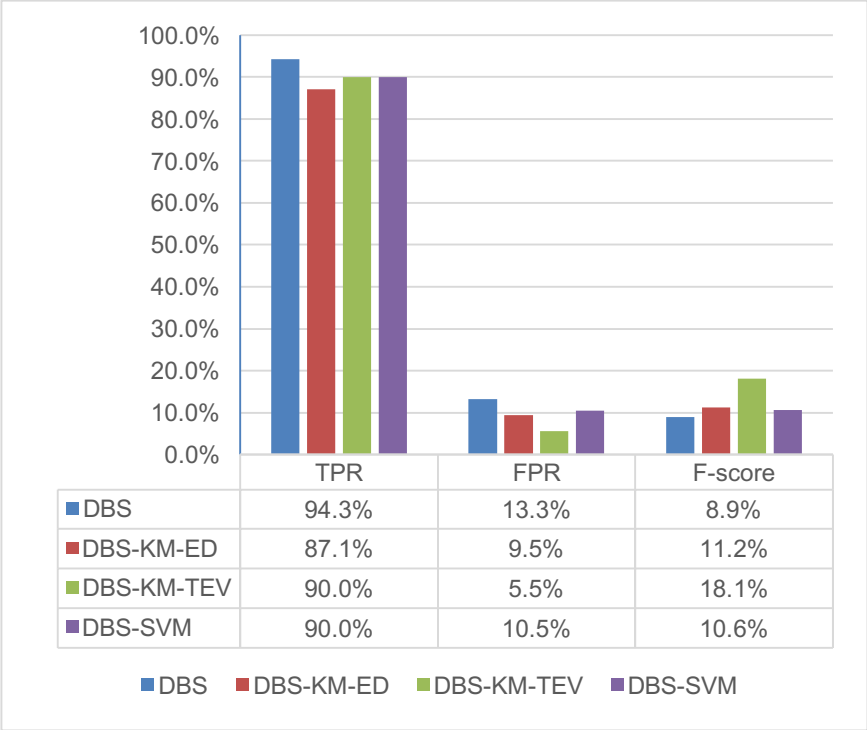
Figura 5.11 ACME-IDE - Desempenho com diferentes valores de τ_2.

A partir do valor τ_1 definido anteriormente, foram realizados testes com valores para τ_2 entre 0,2 e 1. Os resultados obtidos são apresentados na Figura 5.11. Similar ao observado durante a análise do conjunto NSL-KDD, o algoritmo obteve melhores resultados quando τ_2 possui valor igual a τ_1.

No melhor caso, a utilização do mecanismo de encaminhamento aumentou a taxa de detecção em 26%, com aproximadamente 6% das conexões encaminhadas para análise externa. Das 594 conexões encaminhadas, 581 (97,8%) são conexões normais e 13 (2,2%) são conexões relacionadas a ataques.

5.5.4 Comparação dos resultados obtidos

Nesta seção é realizada uma comparação dos resultados obtidos utilizando o método proposto, similar à comparação apresentada na seção 5.3.4. Na Figura 5.12 são apresentadas a taxas de detecção, a taxa de falso positivos e o índice F-score das diferentes abordagens.



	TPR	FPR	F-score
DBS	94.3%	13.3%	8.9%
DBS-KM-ED	87.1%	9.5%	11.2%
DBS-KM-TEV	90.0%	5.5%	18.1%
DBS-SVM	90.0%	10.5%	10.6%

Figura 5.12 ACME-IDE - Comparação entre diferentes abordagens.

O método proposto neste trabalho obteve o melhor valor para o índice F-score. Embora apresente uma taxa de detecção inferior à utilização somente do algoritmo DBSCAN, a combinação proposta possibilitou uma redução de 58% na taxa de falsos positivos, com uma redução de apenas 4% na taxa de detecção. Já a combinação dos algoritmos DBSCAN e *one-class* SVM, diferente do observado no conjunto NSL-KDD, trouxe uma pequena melhora na taxa de falsos positivos, mas com a mesma taxa de detecção da metodologia proposta.

5.6 Considerações finais

Neste capítulo foram apresentados os resultados da utilização de diferentes abordagens utilizando métodos de agrupamento para a detecção de anomalias em redes de computadores. Foram apresentadas quatro abordagens: utilização do algoritmo DBSCAN, duas combinações do algoritmo DBSCAN com o algoritmo k-médias e uma combinação entre o algoritmo DBSCAN e o algoritmo *one-class* SVM. As abordagens foram analisadas utilizando dois conjuntos de dados contendo informações sobre tráfego de rede legítimo e ataques.

Embora a abordagem utilizando o algoritmo DBSCAN tenha apresentado uma taxa de detecção elevada em ambos os conjuntos de dados, esta abordagem apresenta um elevado número de falsos positivos, o que, na prática, torna seu uso inviável. Uma alta taxa de falsos positivos sobrecarrega o trabalho dos analistas de segurança, aumentando as chances de uma ameaça real passar sem ser notada. Portanto, faz-se necessário o uso de técnicas que reduzam o número de falsos positivos.

Através dos resultados obtidos, pode-se concluir que o parâmetro τ_2 não é necessário, uma vez que o modelo proposto obteve os melhores resultados quando τ_2 é igual a τ_1. A partir desta observação, é possível concluir, ainda, que, quando pelo menos uma das características de uma conexão possui uma diferença maior que τ_1 em relação ao centroide, essa conexão já pode ser considerada uma conexão suspeita. Portanto, caso um ataque causa algum distúrbio em ao menos uma das características, é possível realizar uma indicação prévia.

A combinação dos algoritmos DBSCAN e *one-class* SVM também não produziu resultados satisfatórios, reduzindo minimamente a taxa de falsos positivos. Já a combinação dos algoritmos DBSCAN e k-médias proporcionou uma redução significativa na quantidade de falsos positivos, com uma pequena redução na taxa de detecção.

CAPÍTULO 6 - Conclusão

A quantidade de informações sensíveis existente em circulação nos ambientes virtuais torna o monitoramento dos ambientes computacionais uma tarefa obrigatória. Dessa forma, sistemas de detecção de intrusão são temas recorrentes de pesquisas na área de segurança da informação. Neste trabalho é proposto um sistema adaptativo para a detecção de ameaças utilizando técnicas não supervisionadas, visto que a obtenção de dados rotulados com uma descrição dos ataques é algo difícil de ser obtido.

Neste trabalho é proposta uma metodologia capaz de identificar ataques a redes de computadores através da detecção de anomalias presentes no tráfego de rede. A metodologia proposta utiliza uma combinação de técnicas de agrupamento, reduzindo assim a quantidade de falsos positivos quando comparada ao uso dos métodos de forma isolada. O método proposto alcançou uma taxa de detecção de 97,1% e uma taxa de falsos positivos de 6,6%, que pode ser considerado um bom resultado tratando-se de detecção de anomalias utilizando técnicas não supervisionadas. Tais taxas foram obtidas a partir da média dos valores obtidos através de 10 execuções para cada configuração, utilizando a técnica de validação cruzada k*-fold, na* qual o conjunto de dados é separado em k partes de mesmo tamanho, sendo que $k-1$ partes são utilizadas para a construção do modelo e 1 parte é utilizada para validação.

Um dos diferenciais da metodologia proposta é que esta não requer uma etapa de treinamento *off-line*. O sistema tenta identificar possíveis

anomalias presentes nos dados analisados e constrói o perfil normal do ambiente gradativamente, atualizando este modelo constantemente. Dessa maneira, o sistema é capaz de se adaptar a mudanças de comportamento que possam ocorrer ao longo do tempo através de um aprendizado contínuo. Uma desvantagem do trabalho proposto é que este não é capaz de lidar com mudanças abruptas no comportamento. Uma possibilidade para sanar tal deficiência seria desenvolver um módulo de detecção de mudanças abruptas que poderia descartar o modelo agrupamento global atual e deixar que o sistema aprenda o novo comportamento com o passar do tempo.

Os parâmetros necessários para o funcionamento do algoritmo foram testados manualmente via tentativa e erro. Como previsto, diferentes valores apresentam comportamentos completamente diferentes em conjuntos de dados distintos. Como trabalho futuro, é necessário encontrar um método sistemático para refinar os parâmetros do sistema e definição dos limiares de detecção de maneira automática e adaptativa. Os principais parâmetros a serem adaptados são: o limiar utilizado para detecção da anomalia, a janela utilizada durante o pré-processamento e a combinação entre distância máxima e número de elementos para formar grupos no algoritmo DBSCAN.

REFERÊNCIAS

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

AMIRI, F. et al. Mutual information-based feature selection for intrusion detection systems. **Journal of Network and Computer Applications**, v. 34, n. 4, p. 1184–1199, 2011.

BATISTA, M. L. **Análise de Eventos de Segurança em Redes de Computadores Utilizando Detecção de Novidade**. [s.l: s.n.].

BHUYAN, M. H.; BHATTACHARYYA, D. K.; KALITA, J. K. **An effective unsupervised network anomaly detection method**. Proceedings of the International Conference on Advances in Computing, Communications and Informatics - ICACCI '12. **Anais**...: ICACCI '12.New York, New York, USA: ACM Press, 2012Disponível em: <http://doi.acm.org/10.1145/2345396.2345484>

BHUYAN, M. H.; BHATTACHARYYA, D. K.; KALITA, J. K. TOWARDS AN UNSUPERVISED METHOD FOR NETWORK ANOMALY DETECTION IN LARGE DATASETS. **Computing and Informatics**, v. 33, n. 1, p. 1–34, 2014.

CERT.BR. **Estatísticas Mantidas pelo CERT.br**. Disponível em: <http://www.cert.br/stats/>. Acesso em: 6 out. 2014a.

CERT.BR. **Cartilha de Segurança para Internet**. Disponível em: <http://cartilha.cert.br/>. Acesso em: 10 out. 2014b.

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

CHEBROLU, S.; ABRAHAM, A.; THOMAS, J. P. Feature deduction and ensemble design of intrusion detection systems. **Computers & Security**, v. 24, n. 4, p. 295–307, jun. 2005.

CHEN, T. et al. Efficient classification using parallel and scalable compressed

model and its application on intrusion detection. **Expert Systems with Applications**, v. 41, n. 13, p. 5972–5983, out. 2014.

CLAISE, B.; TRAMMELL, B.; AITKEN, P. **Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information**, 2013.

CORRÊA, J. L. **Um modelo de detecção de eventos em redes baseado no rastreamento de fluxos**. [s.l: s.n.].

COSTA, K. et al. Intrusion detection in computer networks using Optimum-Path Forest clustering. **37th Annual IEEE Conference on Local Computer Networks**, p. 128–131, out. 2012.

CRETU, G. F. et al. **Casting out Demons: Sanitizing Training Data for Anomaly Sensors**. 2008 IEEE Symposium on Security and Privacy (sp 2008). **Anais**...IEEE, maio 2008Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4531146>

DENNING, D. E. An intrusion-detection model. **IEEE Transactions on software engineering**, n. 2, p. 222–232, 1987.

ELSHOUSH, H. T.; OSMAN, I. M. Alert correlation in collaborative intelligent intrusion detection systems—A survey. **Applied Soft Computing**, v. 11, n. 7, p. 4349–4365, out. 2011.

ERMAN, J.; ARLITT, M.; MAHANTI, A. Traffic classification using clustering algorithms. **Proceedings of the 2006 SIGCOMM workshop on Mining network data - MineNet '06**, p. 281–286, 2006.

ESKIN, E. et al. A Geometric Framework for Unsupervised Anomaly Detection. In: [s.l: s.n.]. v. 3p. 77–101.

FENG, W. et al. Mining network data for intrusion detection through combining SVMs with ant colony networks. **Future Generation Computer Systems**, v. 37, p. 127–140, jul. 2014.

FREY, B.; DUECK, D. Clustering by passing messages between data points. **science**, v. 315, n. 5814, p. 972–976, 2007.

GAMA, J. **Knowledge Discovery from Data Streams**. [s.l.] Chapman and

Hall/CRC, 2010. v. 20103856

GENTLEMAN, R.; CAREY, V. J. Unsupervised Machine Learning. In: **Bioconductor Case Studies**. New York, NY: Springer New York, 2008. p. 137–157.

HAN-WEI HSIAO; DENG-NENG CHEN; TSUNG JU WU. **Detecting hiding malicious website using network traffic mining approach**. 2010 2nd International Conference on Education Technology and Computer. **Anais**...IEEE, jun. 2010Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5530064>

HU, L.; REN, W.; REN, F. **An Adaptive Anomaly Detection Based on Hierarchical Clustering**. 2009 First International Conference on Information Science and Engineering. **Anais**...IEEE, 2009Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5454947>

IPPOLITI, D.; ZHOU, X. A-GHSOM: An adaptive growing hierarchical self organizing map for network anomaly detection. **Journal of Parallel and Distributed Computing**, v. 72, n. 12, p. 1576–1590, dez. 2012.

LIAO, H.-J. et al. Intrusion detection system: A comprehensive review. **Journal of Network and Computer Applications**, v. 36, n. 1, p. 16–24, jan. 2013.

LICHMAN, M. **UCI Machine Learning Repository**, 2013. Disponível em: <http://archive.ics.uci.edu/ml>

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, 2015.

MARKOU, M.; SINGH, S. Novelty detection: a review—part 1: statistical approaches. **Signal Processing**, v. 83, n. 12, p. 2481–2497, dez. 2003.

MCHUGH, J. Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. **ACM Transactions on Information and System Security (TISSEC)**, v. 3, n. 4, p. 262–294, 2000.

MOHAMAD, I. BIN; USMAN, D. Standardization and its effects on K-means

clustering algorithm. **Research Journal of Applied Sciences, Engineering and Technology**, v. 6, n. 17, p. 3299–3303, 2013.

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

Open Source SECurity. Disponível em: <http://www.ossec.net/>. Acesso em: 15 nov. 2014.

PAXSON, V. Bro: a System for Detecting Network Intruders in Real-Time. **Computer Networks**, v. 31, n. 23–24, p. 2435–2463, 1999.

PHAM, D. T.; DIMOV, S. S.; NGUYEN, C. D. Selection of K in K-means clustering. **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science**, v. 219, n. 1, p. 103–119, 2005.

PORTNOY, L.; ESKIN, E.; STOLFO, S. **Intrusion detection with unlabeled data using clustering**. In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001. **Anais**...2001

ROESCH, M. Snort: Lightweight Intrusion Detection for Networks. **LISA**, v. 99, p. 229–238, 1999.

SCHÖLKOPF, B. et al. Estimating the support of a high-dimensional distribution. **Neural computation**, v. 13, n. 7, p. 1443–1471, 2001.

SILVA, J. A et al. Data stream clustering. **ACM Computing Surveys**, v. 46, n. 1, p. 1–31, 1 out. 2013.

SPINOSA, E. J.; DE LEON F. DE CARVALHO, A. P.; GAMA, J. **Cluster-based novel concept detection in data streams applied to intrusion detection in computer networks**. Proceedings of the 2008 ACM symposium on Applied computing - SAC ’08. **Anais**...: SAC ’08.New York, New York, USA: ACM Press, 2008Disponível em: <http://doi.acm.org/10.1145/1363686.1363912>

Suricata. Disponível em: <http://suricata-ids.org/>. Acesso em: 25 nov. 2014.

SUTHAHARAN, S. Big Data Classification: Problems and challenges in network intrusion prediction with machine learning. **Big Data Analytics**

workshop, in conjunction with …, 2013.

TAVALLAEE, M. et al. **A detailed analysis of the KDD CUP 99 data set**. 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. **Anais**...IEEE, jul. 2009Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5356528>

THAKRAN, Y.; TOSHNIWAL, D. **Unsupervised outlier detection in streaming data using weighted clustering**. 2012 12th International Conference on Intelligent Systems Design and Applications (ISDA). **Anais**...IEEE, nov. 2012Disponível em: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6416666>

TRAMMELL, B. H.; BOSCHI, E. **Bidirectional Flow Export Using IP Flow Information Export (IPFIX)**. article. Disponível em: <https://tools.ietf.org/html/rfc5103>. Acesso em: 22 ago. 2014.

WANG, W. et al. Autonomic intrusion detection: Adaptively detecting anomalies over unlabeled audit data streams in computer networks. **Knowledge-Based Systems**, v. 70, p. 103–117, nov. 2014.

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.

ZANDER, S.; NGUYEN, T.; ARMITAGE, G. **Automated traffic classification and application identification using machine learning**. The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)l. **Anais**...IEEE, 2005Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1550864>

ZHANG, X.; FURTLEHNER, C.; SEBAG, M. Data Streaming with Affinity Propagation. In: **Machine Learning and Knowledge Discovery in Databases**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 628–643.

ZHONG, S.; KHOSHGOFTAAR, T. M.; SELIYA, N. CLUSTERING-BASED NETWORK INTRUSION DETECTION. **International Journal of Reliability, Quality and Safety Engineering**, v. 14, n. 2, p. 169–187, abr. 2007.

ZIKOPOULOS, P.; EATON, C. **Understanding big data: Analytics for enterprise class hadoop and streaming data**. [s.l.] McGraw-Hill Osborne Media, 2011.