UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

WEVERTON CARVALHO

ALGORITMO DE CÉLULAS DENDRITÍCAS ATRAVÉS DO MODELO DO PERIGO PARA UM SISTEMA DE DETECÇÃO DE INTRUSOS BASEADO EM ANOMALIAS

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2015

WEVERTON CARVALHO

ALGORITMO DE CÉLULAS DENDRITÍCAS ATRAVÉS DO MODELO DO PERIGO PARA UM SISTEMA DE DETECÇÃO DE INTRUSOS BASEADO EM ANOMALIAS

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientadora: Profª Drª Tânia Lúcia Monteiro

PONTA GROSSA

2015

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa
Diretoria de Graduação e Educação Profissional
Departamento Acadêmico de Informática
Bacharelado em Ciência da Computação

TERMO DE APROVAÇÃO

ALGORITMO DE CÉLULAS DENDRITÍCAS ATRAVÉS DO MODELO DO PERIGO PARA UM SISTEMA DE DETECÇÃO DE INTRUSOS BASEADO EM ANOMALIAS

por

WEVERTON CARVALHO

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 12 de Novembro de 2015 como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____
Profª Drª Tânia Lúcia Monteiro
Prof.(a) Orientador(a)

_____
Prof MSc Rogério Ranthum
Membro titular

_____
Prof MSc Vinícius Camargo Andrade
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

Dedico este trabalho a Deus e aos meus pais, que com muito apoio e carinho, não mediram esforços para que eu chegasse até aqui.

AGRADECIMENTOS

Primeiramente agradeço a Deus, que permitiu que tudo isso acontecesse, Por me amparar não apenas nestes anos como universitário, mas ao longo de minha vida, nos mais diversos momentos.

Aos meus pais, minha irmã, pelo apoio e por tudo que sempre fizeram por mim, pela simplicidade, exemplo, carinho e compreensão. Além do grande incentivo nas horas mais difíceis, de desânimo e cansaço.

A minha namorada pelos incentivos e compreensão nos momentos de ausência.

A toda minha família, avós, avôs, tios e primos, por compartilharem diversos momentos e conselhos importantes em minha vida desde o início dessa caminhada.

À minha orientadora, Profª Drª Tânia Lúcia Monteiro, que acreditou em minha proposta, ouviu pacientemente minhas considerações partilhando suas idéias, conhecimento e experiências. Quero expressar o meu reconhecimento e admiração pela sua competência profissional como professora orientadora.

Aos meus amigos, que considero minha segunda família. Obrigado pela paciência, pelos momentos incríveis, tristes e alegres. Esta jornada acadêmica não seria o mesmo sem esses momentos juntos.

Aos professores que proporcionaram o conhecimento, que me ajudou a crescer tanto profissionalmente como pessoalmente. Conhecimento este não apenas racional e lógico más crítico e de caráter. Obrigado pelas horas dedicadas ao meu ensino, pela atenção com a qual sempre fui tratado. Sem nominá-los deixo aqui os meus sinceros e eternos agradecimentos.

RESUMO

CARVALHO, Weverton. Algoritmo de Células Dendritícas Através do Modelo do Perigo para um Sistema de Detecção de Intrusos Baseado em Anomalias. 2015. 55 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2015.

Sistemas de detecção de intrusos referem-se a meios de descobrir se uma rede está tendo acessos indevidos, que podem indicar uma anomalia causada por uma má configuração de software, mau uso do mesmo, ou ação de um hacker. Esses sistemas são capazes de analisar o tráfego em uma rede, atividades que acontecem em um determinado computador e decidir se as mesmas constituem-se em ataques ou simples ações rotineiras. A motivação para este trabalho são os problemas que os algoritmos classificadores atuais vêm proporcionando. Os sistemas de detecção de intrusão possuem alguns problemas como falsos positivos, falsos negativos, erros de subversão e limitadores físicos na infraestrutura. Estes problemas ocasionam uma grande taxa de erro em decisões tomadas. Neste trabalho será proposto a implementação de um algoritmo baseado em técnicas de inteligência artificial para diminuir as taxas de erros em detecções de intrusão.

Palavras-chaves: Sistemas de Detecção de Intrusos. Sistemas Imunes Artificiais. Células Dendríticas. Teoria do Perigo.

ABSTRACT

CARVALHO, Weverton. Dendritic Cell Algorithm Through of Danger Model for Intrusion Detection Systems Based on Anomaly. 2015. 55 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2015.

Intrusion detection systems are techniques to find if a network is having unauthorized access that can be an anomaly, caused by wrong software configuration or an inappropriate use or hacker attack. This systems are able to analyze network traffic, some activities occurring in a computer and to decide if the situations are simple events or some hacker attack. The reason to propose this final paper is due to the problems found about current classification algorithms. Intrusion detection systems have some problems like false positive, false negative, subversion errors and physical constraints on infrastructure. These problems are responsible to cause a large error of rate decision. This final paper propose a implementation of an algorithm supported by artificial intelligence techniques to decrease the error rate by intrusion detection algorithms.

Keywords: Intrusion Detection Systems. Artificial Immune Systems. Dendritic Cell. Danger Theory.

LISTA DE FIGURAS

LISTA DE GRÁFICOS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE SIGLAS

APC	Célula apresentadora de antígenos (*Antigen Presenting Cell*)
DARPA	*Defense Advanced Research Projects Agency*
DCA	Algoritmo das Células Dendríticas (*Dendritic Cell Algorithm*)
DoS	Negação de serviço (*Denial of Service*)
DP	Desvio Padrão
FN	Falso Negativo
FP	Falso Positivo
GB	*Giga Bytes*
ICMP	Protocolo de mensagens de controle da internet (*Internet Control Message Protocol*)
IDS	Sistema detector de intrusos (*Intrusion Detection System*)
IP	Protocolo da Internet (*Internet Protocol*)
IRC	*Internet Relay Chat*
PAMP	Padrão molecular associado ao patógeno (*Pathogen-associated molecular pattern*)
RAM	Memória de Acesso Aleatório (*Random Access Memory*)
SI	Sistemas Imunes
SIA	Sistemas Imunes Artificiais
Sm	Semi Madura
Sp	Sinal de Perigo
SVM	*Support vector machine*

SUMÁRIO

1 INTRODUÇÃO

Com o rápido desenvolvimento da tecnologia baseada na *internet* e a dependência de negócios críticos aos sistemas de informação, novos domínios de aplicação em redes de computadores vêm surgindo (STALLINGS, 2006). Devido a este fato, ocorreu a ampliação dos mecanismos de segurança em redes computacionais, ocasionando a evolução de programas maliciosos, denominados *malwares*, que estão cada vez mais estruturados, ou seja, estão ficando cada vez mais complexos e difíceis de serem detectados.

Os mecanismos de prevenção, tais como, criptografia e autenticação, são a primeira linha de defesa em uma rede, garantindo alguns princípios de segurança como confidencialidade e integridade (STALLINGS, 2006). Porém, quando estas medidas não são suficientes para lidar com todos os tipos de ataque, faz-se necessário um segundo mecanismo de segurança, os Sistemas de Detecção de Intrusos (DEBAR; DACIER; WESPI, 2000).

O conceito de *Intrusion Detection System (IDS)* surgiu nos anos 80 em estudos do *Stanford Research Institute*. Conhecido como *Project 6169 - Statistical Techniques Development For An Audit Trail System*, o projeto utilizava um algoritmo de alta velocidade que analisava os usuários com base nos seus perfis de comportamento.

De forma geral, estes sistemas referem-se a meios de descobrir se uma rede está tendo acessos indevidos que podem indicar uma anomalia causada por uma má configuração de *software*, mal uso do mesmo, ou ação de um atacante mal intencionado. Eles são capazes de analisar o tráfego em uma rede, atividades que acontecem em um determinado computador e decidir se os mesmos eventos são ataques ou simples ações rotineiras por meio de determinados métodos.

Há diversos métodos de detecção de anomalias no tráfego de rede, como métodos baseados em análise estatística (SAMAAN; KARMOUCH, 2008), estatística bayesiana (LIU, 2008), algoritmos de agrupamento (LI; LEE, 2003), lógica *fuzzy* (YAO; ZHITANG; SHUYU, 2006), algoritmos genéticos (SELVAKANI; RAJESH, 2007) e sistemas imunológicos artificiais (GUANGMIN, 2008) (PERLIN; NUNES; KOZAKEVICIUS, 2011).

A motivação para este trabalho se deve aos resultados apresentados na literatura, pelos algoritmos classificadores, que tratam os sistemas de detecção de intrusos.

Estes sistemas possuem alguns problemas como falsos positivos, falsos negativos, erros de subversão e limitadores físicos na infraestrutura, que ocasionam uma grande taxa de erro em decisões tomadas e milhões em prejuízos para empresas (Ponemon; IBM, 2014).

Vários estudos ao longo dos últimos anos foram feitos na área de sistemas imunes para resolver estes problemas, em particular a teoria do perigo introduzida por Polly Matzinger (MATZINGER,1994) que inspirou pesquisadores a desenvolverem sistemas computacionais para a segurança em redes (AICKELIN; CAYZER, 2002).

Um dos primeiros algoritmos a utilizarem a teoria do perigo foi proposto em Greensmith, o qual implementou-se um algoritmo para sistemas detecção de intrusos denominado algoritmo das células dendritícas para detecção do uso malicioso de *ping scan* (GREENSMITH, 2007).

Neste trabalho será proposto à implementação do algoritmo das células dendritícas para diminuir as taxas de erros para a detecção de vários tipos de ataques. Para que se consiga chegar o mais próximo da realidade será utilizado uma base de dados real a KDD Cup 99 (KDD Cup 1999 Data, 1999). Os dados da KDD Cup 99 de treinamento brutos têm em torno de 4 GB obtidos através de TCP *dump* por sete semanas no tráfego de rede da *DARPA*. Com isso, os dados foram comprimidos em cerca de 4 milhões de registros de exemplo (conexões) e estão disponíveis na página do KDD Cup 99 (Souza; Silva, 2008).

Para a validação dos testes utilizou-se a suite WEKA (*Waikato Enviroment for Knowledge Analysis*), formada por um conjunto de implementações de algoritmos de diversas técnicas de Mineração de Dados (UNIVERSITY OF WAIKATO, 2010).

Dentre essas técnicas selecionou-se alguns algoritmos de acordo com Garg (GARG; KHURANA, 2014) para a comparação dos mesmo com o algoritmo implementado.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

- Comparar o algoritmo das células dendríticas com os algoritmos classificadores disponíveis na literatura, tendo como base de comparação suas respectivas performances.

1.1.2 Objetivos Específicos

- Explorar a base de dados DARPA KDDCUP 99 (KDD Cup 1999 Data, 1999);
- Implementar o algoritmo de Células Dendríticas através modelo do perigo;
- Análise e seleção dos algoritmos classificadores;
- Utilizar WEKA para a comparação dos algoritmos classificadores.

1.2 JUSTIFICATIVA

Com a grande taxa de falsos positivos e falsos negativos gerados, os sistemas de detecção podem levar o usuário a tomar decisões erradas, o que pode acarretar prejuízos para empresas. O algoritmo a ser desenvolvido terá como objetivo a diminuição desses erros em sistemas de detecção de intrusos, para que haja uma maior confiabilidade durante uma tomada de decisão pelo usuário do mesmo.

A utilização do algoritmo de células dendríticas e modelo do perigo se da pela alta aplicabilidade nas áreas de segurança computacional. Além de serem aplicados na detecção, possuem ótimos resultados quando se trata de falsos positivos e falsos negativos (SILVA, 2009). A base de dados DARPA KDDCUP 99, será usada por ser uma base retirada de um ambiente real.

2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentados conceitos envolvendo Segurança Computacional, Sistemas de Detecção de Intrusos, Sistemas Imunes, assim como o conceito da Teoria do perigo, algoritmo das Células Dendríticas e a base de dados KDDCUP'99.

2.1 SEGURANÇA COMPUTACIONAL

Com o passar dos anos a tecnologia de redes de computadores vem sendo aplicada em grande escala pela sua facilidade de implantação, tanto em ambientes domésticos, como em corporativos. O uso de técnicas de segurança se tornou algo indispensável, com o avanço de serviços críticos baseados em *internet*, onde se tem transações com dados confidenciais de seus usuários. Nesta seção serão apresentados alguns conceitos relacionados à segurança computacional.

2.1.1 Conceito

Conceito de segurança é padronizado pela norma NBR ISO/IEC 27002 (ABNT, 2005). Onde segurança da informação é definida como a preservação da confidencialidade, da integridade e da disponibilidade da informação; adicionalmente outras propriedades, tais como autenticidade, responsabilidade, não repudio e confiabilidade, podem também estar envolvidas. As propriedades anteriormente citadas podem ser caracterizadas como:

- Confidencialidade: garantia de proteção contra leitura ou cópia por usuário não autorizado.

- Integridade: significa garantir a proteção de qualquer informação ou sistema contra modificações ou remoções sem a permissão explícita de seu proprietário ou do sistema que a disponibiliza.

- Disponibilidade: garantia de proteção de qualquer sistema ou serviço contra a degradação e/ou indisponibilidade sem autorização. Implica se a informação ou serviço estará sempre disponível.

- Autenticidade: serve para atestar a validade das informações e sua origem, garantindo a identidade de entidades que interagem no sistema.

- Responsabilidade: definir a partir de uma política, o modo de como e onde cada ação poderá ser executada ou não.

- Não repúdio: possibilidade de identificação de autoria de determinada ação sem equívocos, fornecendo prova irrefutável da realização de uma ação específica por parte de algum usuário.

- Confiabilidade: capacidade do sistema manter o seu funcionamento em circunstancias normais e hostis.

A não aplicação ou utilização errada da segurança pode comprometer o sistema computacional de forma a permitir que ameaças possam obter acesso à informações restritas.

2.1.2 Ameaças

As ameaças ocorrem frequentemente tentando comprometer ambientes computacionais por meio de vulnerabilidades. Estas podem ser desencadeadas de forma acidental, maliciosa, de modo externo ou interno, tendo por finalidade desde o vandalismo até a espionagem corporativa. Proporcionando ao atacante a intrusão por meio de ataques.

Os desafios para se conter essas ameaças são grandes, principalmente em meios corporativos, pois empresas estão suscetíveis a várias influências tais como: medidas econômicas adotadas pelo governo, política, mercado financeiro, cambial ou mesmo as mais internas como a setorial, o técnico, pessoal, e físico.

Comparativamente, são muitos os fatores que influenciam as medidas de segurança da informação de uma organização.

Com o mundo conectado a *internet*, a principal ameaça se da pela violação de dados. Um estudo de caso desenvolvido pela IBM e Ponemom mostra que o custo total médio de uma violação de dados para as empresas participantes cresceu 15%, chegando a US$3,5 milhões. O custo médio pago pela perda ou roubo de um registro que contém informações sensíveis ou confidenciais subiu mais de 9%, ou seja, de US$136 em 2013 para US$145 em 2014. Sendo que as conclusões sugerem que organizações da Índia e do Brasil estão mais propensas a ter uma violação de dados envolvendo pelo menos 10.000 registros. No Brasil, o custo médio total para uma empresa foi de US$1,61 milhão (PONEMON; IBM, 2014).

Tendo por objetivo a amenização dessas ameaças o sistema de detecção de intrusos é um dos principais meios para se identificar esses tipos de ataques.

2.1.3 Ataques

Ocorre quando uma ameaça é realizada, onde se tem a violação das políticas de segurança e por consequência o comprometimento do ambiente computacional. Esses ataques vêm crescendo exponencialmente nos últimos anos, como pode-se observar no Gráfico 1.

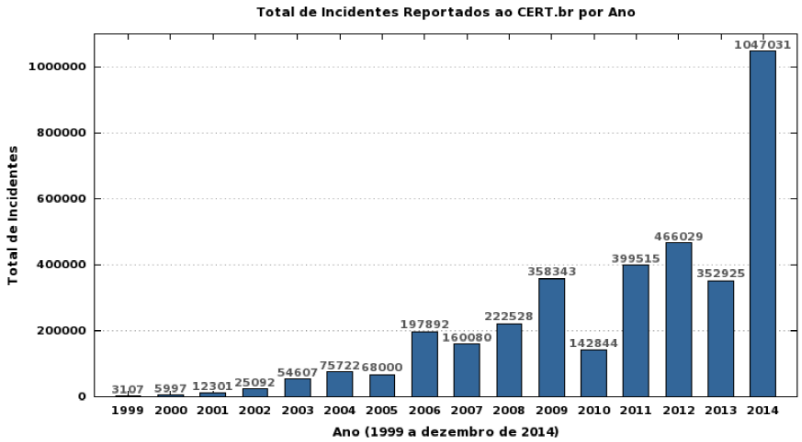

Gráfico 1- Incidentes reportados ao Centro de Estudos, Respostas e Tratamento de Incidentes de Segurança no Brasil.
Fonte: CERT.br (2015)

Os tipos de ataques são classificados de acordo com o Quadro 1.

Nome do ataque	**Sintoma**	**Descrição**	**Notas**
Boink (semelhante ao *Bonk* , *Teardrope New Tear/Tear2), hack.*	Embargo do sistema.	Ataque de fragmento mal intencionado.	Envia fragmentos de pacote incorretos que não podem ser corretamente remontados, fazendo com que o sistema falhe.
DoS (Denial of Service).	Falta de acesso a recursos e serviços.	Ataques de negação de serviço Travam os recursos do sistema, utilizando-os de forma desnecessária de modo a não permitir o uso do serviço oferecido.	Exemplos incluem *floods* (que absorvem a largura de banda e CPU) e desconexões (impedem que se acesse host ou redes).
Floods, um ataque *DoS*.	Não aplicável.	Grandes quantidades de pacotes inúteis de *ICMP* (geralmente) ou *UDP*.	Trava o sistema, fazendo com que responda aos *floods*.
ICMP flooding (ping flood), ataque *DoS*.	Perda de banda (respostas lentas a partir da *Internet*) e tempo de resposta ruim no ambiente de trabalho.	*Flood ICMP* (*ping*) pedidos que travam o sistema em *looping*, fazendo com que responda ao tráfego de lixo. Isto é análogo a desperdiçar o seu tempo atendendo a campainhas de porta infinitas que não fazem nada.	Limita o tempo de CPU e desperdiça sua banda com o tráfego de lixo. Por exemplo, "*Pingexploit*" normalmente ataca sistemas *Unix* com fragmentos de pacotes *ICMP* de grandes tamanho.
Identification *flooding* (*Identd*), ataque *DoS*.	Perda de banda (respostas lentas a partir da *Internet*) e tempo de resposta ruim no ambiente de trabalho.	Semelhante ao *ICMP*, mas solicita informações de seu sistema (porta TCP 113).	Retarda a *CPU* Muito frequentemente, deixando-a lenta (ainda mais do que um *ICMP flood*), uma vez que as respostas de identificação levam mais tempo do que o

			ICMP para gerar respostas.
Jolt (SSping, Ice-Nuke), hack.	Embargo do Sistema.	Pacotes fragmentados em grandes dimensões, causando sobrecarga no sistema.	Sistema pára de funcionar e deve ser reinicializado.
Land, hack.	Embargo do Sistema, forçando a reinicialização manual.	Tentativa de *spoofing* que estabelece conexão o *TCP* / *IP* de você para você . Este pedido *SYN* obriga o sistema a ligar-se a si próprio , causando sobrecarga.	O sistema atacado tenta se conectar a si mesmo, causando sobrecarga.
Hack.	Não aplicável.	Um aplicativo ou pacote que explora uma fraqueza de um sistema operacional, aplicação ou protocolo.	Resultados variados. Incluem-se nos exemplos o *smurf*, *teardrop*, *land, newtear, puke, ssping, jolt*, etc.
Pong, hack.	Perda da largura de banda (respostas lentas a partir da *Internet*) e tempo de resposta ruim no ambiente de trabalho.	Grande fluxo de pacotes *ICMP* falsos geralmente muda se e falsifica o endereço de origem em todos os pacotes	Necessário reiniciar para resolver.
Puke, hack.	Desconexão do servidor (normalmente *IRC*).	Falsifica o *ICMP* com erro de destino inacessível. Isso força a se desconectar de um servidor.	Geralmente precedida por uma varredura de portas ICMP onde "*ping* "são enviados à um sistema para encontrar portas vulneráveis que estão sendo usadas para conexão de um servidor.
Scan, técnica genérica	Deixa o sistema lento.	Um progressivo e	Normalmente

e um ataque *DoS*.		sistemático teste de portas, para determinar as portas abertas. Este ataque pode consumir os recursos do sistema uma vez que o alvo geralmente esta mudando. Para a prevenção se requer um firewall apropriado ou bloco multi- porta.	utilizado antes de um *hack* para encontrar lugares vulneráveis à ataques. Isto é considerado uma forma brutal de ataque e não é tão eficaz como outras técnicas de *hack* para travar recursos . Ele geralmente precede uma forma de ataque "elegante ".
Smurf, hack.	Muito eficaz para exaustão de *CPU* semelhante ao ataque de *flood*. Sistema aparentemente fica sobrecarregado.	Falsifica-se pacotes *ICMP* solicitando uma resposta, provocando o retorno de múltiplas respostas.	A forma de *flood* que é muito perigoso, já que ele pode ficar em um efeito de "muitos- para -um", travando. Muitos ciclos de CPU, para relativamente enviar poucos pacotes.
Spoofing (*IPspoof*).	Não aplicável.	Um estilo de ataque de máscara que faz com que o tráfego pareça vir de um alvo legitimo ou que faz a estrutura do ataque parecer inocente, para que não seja detectado.	Ataques são particularmente desagradáveis porque *hacks* são ilegais na maioria dos países e sujeitos a acusações.
Unreachable (dest_unreach) - ataque DoS.	Mensagem de "Destino inacessível" e desconexão do servidor.	Existem duas formas desse ataque: - cliente inacessível e o servidor inacessível. O ataque de servidor inacessível envia uma mensagem *ICMP* para o sistema	Não aplicável.

		enganando-o. – fazendo o sistema pensar que o seu tráfego não pode mais alcançar o servidor , logo ele desiste. O cliente fica de forma inalcançável, fazendo a mesma coisa com o servidor em relação ao seu sistema.	
WinNuke, *hack* e ataque de negação de serviço, porem não é um *flood*.	Perda de recursos da rede.	*Envia OOB (Outof-Dados BAND)* para a porta 139 e explora sistemas Win 3.11, Win95 , Win NT 3.51 e Win NT 4.0.	Não trava o sistema , mas causa uma exceção fatal exigindo a reinicialização para recuperar a conectividade (*Internet*) por meio do *TCP/IP*.

Quadro 1 – Tipos de ataques.
Fonte: (TJADEN, 2001) (Tradução do autor).

Para evitar uma possível invasão, por meio de ataques que venham a explorar vulnerabilidades no sistema, necessita-se de ferramentas que possam combater essas ações. Um dos meios de defesa é dominado sistema de detecção de intrusos, em que se tem um monitoramento do ambiente computacional para determinar se as ações são normais ou possíveis invasões.

2.2 SISTEMAS DE DETECÇÃO DE INTRUSÃO

É um suporte de novas tecnologias de segurança que monitoram o sistema de rede sem afetar o seu desempenho interno e externo para prevenir ataques e uso indevido (ZHU et al., 2008). Pode ser visto como mais uma ferramenta para reforçar a política de segurança da informação de uma empresa. Possibilita a detecção e o bloqueio de ataques antes que eles obtenham sucesso. Para que possam monitorar

possíveis ataques usa-se os métodos de detecção, para determinar se as ações ocorridas no sistema são normais ou intrusivas.

2.2.1 Métodos de Detecção

Através destes métodos é efetuado o desenvolvimento de um componente importante na construção de sistemas de detecção de intrusos, pois permitem a ação e implementação do principal método de solução. Neles se aplicam as formas de detecção, podendo ser uma assinatura onde se tem um monitoramento baseado em padrões de ataques conhecidos. Também pode-se utilizar detecção por anomalias através de métodos estatísticos ou híbridos que aplicam ambas as formas de detecção.

2.2.1.1 Método de detecção por anomalia

Estas técnicas visam detectar intrusos com base nas anomalias do tráfego de rede (GARCIA-TEODORO et al., 2009) (SABAHI; MOVAGHAR, 2008), tais como: a alta latência da rede, elevados volumes de tráfego, o tráfego em portas incomuns e comportamento anormal do sistema, que poderiam indicar a presença de atividades maliciosas na rede (FEILY; SHAHRESTANI; RAMADASS, 2009) (SZYMCZYK,2009). No entanto, possui uma grande desvantagem que pode gerar alguns problemas. Em (SILVA, 2009) classifica-se como:

- falso positivo - quando uma ação é classificada como uma possível intrusão, mas se trata de uma ação normal.

- falso negativo - quando ocorre uma ação intrusiva porém ela é classificada como uma ação normal.

- subversão - situação ocorrida quando uma operação do sistema detector de intrusos é modificada para forçar ocorrências de falsos negativos.

A Figura 1 mostra um sistema típico de detecção de anomalias.

Figura 1- Sistema típico de detecção de anomalias.
Fonte: SUNDARAM (2009)

A Figura 2 apresenta graficamente o resultado da técnica aplicada em um sistema de detecção de anomalias.

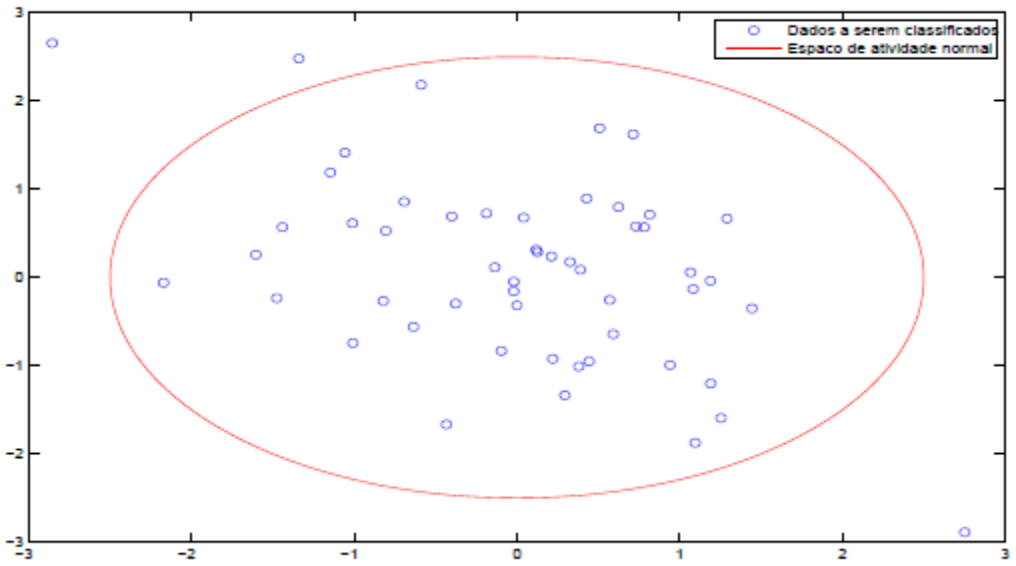

Figura 2 - Representação de uma detecção de anomalias em um plano cartesiano
Fonte: SILVA (2009).

A Figura 2 apresenta um cenário onde há vários dados a serem classificados. Estes dados podem ser classificados como normais ou anormais, para isto usa-se algum tipo de limiar que está representado pela elipse no plano cartesiano. Logo os dados que estão dentro do limiar são classificados como normais, porém os dados que saem do padrão e estão fora do limiar são classificados como anômalos.

Dentre os métodos de detecção baseado em anomalia, destaca-se o uso e aplicação de teorias baseadas em sistemas imunes.

2.2.1.2 Método de detecção por assinatura

Baseia-se em comportamentos intrusivos já conhecidos que se assemelham a eventos intrusivos. Esses comportamentos são chamados de assinatura de intrusão, elas são relacionadas e comparadas aos eventos do sistema para uma possível detecção de intruso.

Estes sistemas possuem limitações de "conhecimento" de rede, protocolos e compreensão de estado de comunicações complexas. São incapazes de armazenar pedidos anteriores enquanto processam o pedido atual, o que impede a detecção de ataques com eventos múltiplos, caso nenhum dos eventos contenha uma indicação clara de um ataque (SILVA, 2009). Quando se tem ataques desconhecidos que ainda não foram divulgados publicamente ou descobertos, este método se torna ineficiente.

A Figura 3 ilustra um sistema típico de detecção por assinatura.

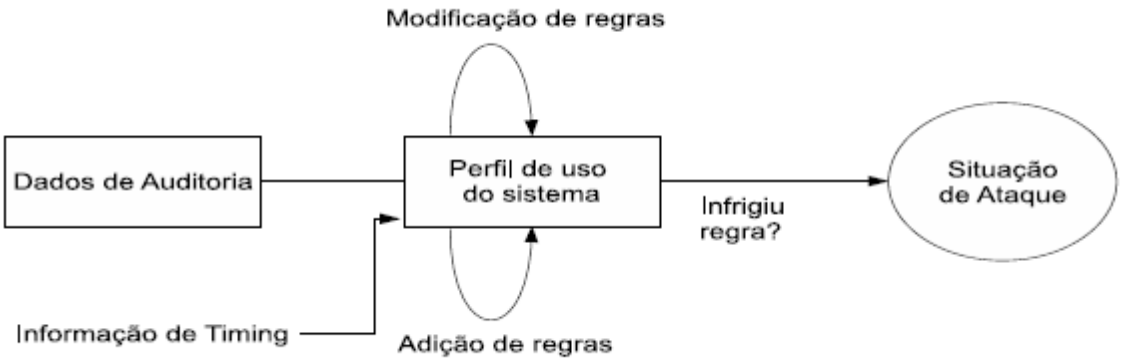

Figura 3 - Sistema típico de detecção de assinaturas
Fonte: SUNDARAM (2009).

A figura 4 apresenta uma representação gráfica de uma detecção de assinaturas.

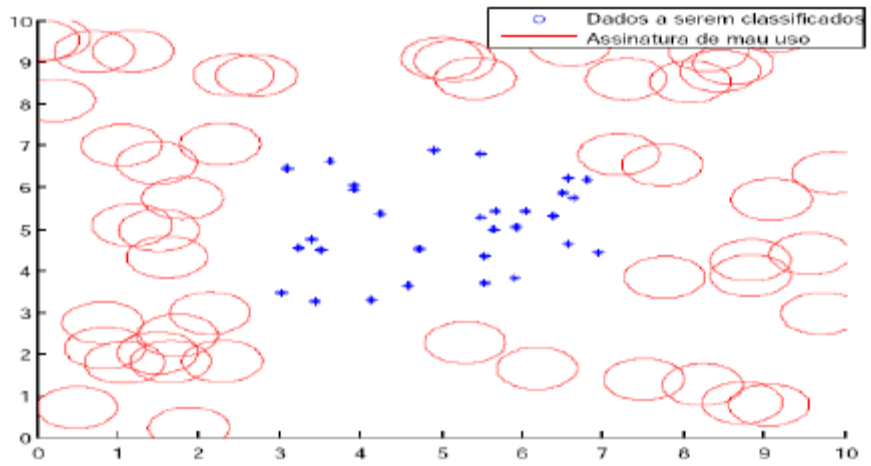

Figura 4 - Representação de uma detecção de assinaturas em um plano cartesiano. Fonte: SILVA (2009).

Representado na Figura 4, este método apresenta vários dados a serem classificados, podendo ser interpretados como uma ação normal ou maliciosa. Para classificar estes dados utilizam-se regras que definem o mau uso, ou seja, quando uma regra é quebrada caracteriza-se com uma ação maliciosa. Estas assinaturas de mau uso podem ser observadas como as elipses, sendo os dados a serem classificados como os círculos azuis.

2.2.1.3Método de detecção híbrido

Esta abordagem utiliza a detecção por anomalia e assinatura, podendo ser adequada para casos distintos. Incorporando os dois métodos pode se ter uma maior eficiência, levando em conta a quantidade e derivações de ataques que existem.

2.3 Sistemas Imunes – SI –

O estudo de sistemas imunológicos esta crescendo nos últimos anos, principalmente por cientistas da computação, matemáticos, engenheiros dentre outros. Estes pesquisadores estão particularmente interessados na capacidade deste sistema para aplicações artificiais. Nesta seção será abordada uma breve introdução sobre sistema imunológico natural.

2.3.1 Sistema Imunológico Natural

O sistema imunológico é um dos sistemas biológicos mais complexos de que se tem conhecimento. Apesar da especificação da sua função ser a princípio bastante simples – detectar e eliminar qualquer organismo estranho – a execução desta tarefa não é trivial. A vida sempre está em constante equilíbrio, um simples acidente com uma farpa que penetre na pele, causa instabilidade no organismo do indivíduo. Logo o ser vivo reconhece que esta sofrendo agressões, o tipo de agressor e a melhor forma de combatê-lo, ou sucumbe aos danos causados pelo invasor. A capacidade do organismo encontrar e destruir um invasor é um processo denominado resposta imunológica (FIGUEREDO; BERNARDINO; BARBOSA, 2013).

2.3.2 Células Dendríticas

O Sistema Imune possui meios de se proteger e se adaptar a possíveis danos sofridos ao indivíduo. Um desses meios é a imunidade natural (nativa ou inata), este mecanismo está presente no indivíduo desde o seu nascimento. Outro meio é a imunidade adaptativa, sendo este mecanismo ativado quando há exposição de fatores estranhos no organismo.

Quando temos a invasão por um tecido estranho, as células brancas são as responsáveis por efetuar a contra medida. Dentre as células brancas se destacam as denominadas dendríticas.

O papel das células dendríticas é capturar e apresentar o antígeno aos linfócitos, a fim de que ele seja reconhecido e combatido. Existem em sua superfície receptores capazes de reconhecer estruturas comuns a diversos tipos de antígenos. Assim, quando estes receptores encontram um antígeno, a célula dendrítica é estimulada a englobá-lo e degradá-lo. Ao ingerir o antígeno, a célula se ativa, amadurecendo em uma célula apresentadora de antígenos (APC) e migra até o linfonodo mais próximo. Já amadurecida, a célula é capaz de ativar os linfócitos (células brancas adaptativas) antígenos-específicos que determinam quando e como o sistema imune deverá responder aos agentes infecciosos. Pode-se dizer que os linfonodos são órgãos que atuam como “centros de inteligência” do SI, onde é

procurado qual o linfócito mais apto a combater o antígeno apresentado pela APC (como as células dendríticas) e onde se dá o início à resposta ao agressor (FIGUEREDO; BERNARDINO; BARBOSA, 2013).

Linfócitos são células imunocompetentes recirculantes responsáveis pela imunidade específica. Estas células realizam um intercurso contínuo entre corrente sanguínea e linfa, e em seguida no sentido inverso, num processo chamado recirculação(CORMACK, 1991). Estas reações e medidas, inspiraram pesquisadores a fazer utilização desses conhecimentos em aplicações e problemas do cotidiano, logo surgiram os sistemas imunes artificiais.

2.3.3 Sistemas Imunes Artificiais – SIA –

SIA são complexos, baseados em sistemas imunológicos naturais que são compostos de moléculas e órgãos capazes de executar diversas funções. Dentre elas se destacam o aprendizado, reconhecimento de padrões, aquisição de memória, geração de diversidade, tolerância a ruídos, generalização, detecção distribuída e otimização.

Sistemas artificiais surgiram na tentativa de aplicar e modelar princípios imunológicos, no desenvolvimento de ferramentas computacionais. Esta aplicação já vem sendo efetuada nas mais diversas áreas como aprendizagem de máquina, detecção de falhas e anomalias, robótica, reconhecimento de padrões e segurança computacional (FIGUEREDO; BERNARDINO; BARBOSA, 2013). Neste trabalho serão utilizadas duas técnicas importantes baseadas em sistemas imunes, sendo elas a teoria do perigo e o algoritmo das células dendríticas.

2.3.4 Teoria do Perigo

Proposto por Polly Matzinger em 1994 (MATZINGER, 1994), enfatiza o papel do sistema imunológico inato, a orientar as respostas imunes adaptativas. No entanto, ao contrário de detectar sinais exógenos, a Teoria do Perigo repousa na detecção de sinais endógenos. Sinais endógenos de perigo surgiram como resultado de uma lesão ou estresse para as próprias células do tecido. O ponto crucial da teoria é que os únicos patógenos detectados são os que induzem a necrose e

causam dano real para o tecido hospedeiro. O dano pode ser causado pela invasão de microrganismos ou por meio de defeitos no tecido do hospedeiro ou células imunes inatas (GREENSMITH,2007).

Em (SILVA, 2009), o autor explica o modelo do Perigo (MATZINGER, 1994), onde as células que apresentam antígenos (*APC*) são ativadas por células que estão sofrendo danos. Os danos funcionam como alarmes ao organismo, de forma a ativar as *APCs* através das células expostas a agentes nocivos ao organismo, como consequência do dano à célula. Neste caso, o sistema imune estaria reagindo contra uma detecção de perigo, em vez de reagir contra padrões desconhecidos ao organismo. O sistema imuno-biológico realizará a defesa do organismo enquanto houver uma situação de perigo, onde algumas células estão sendo atacadas. Analisando intuitivamente o processo de detecção de intrusos em redes de computadores, o problema da intrusão se encaixa no escopo da imunologia. Considerando as intrusões como os agentes patógenos, as características da intrusão são analisadas como os antígenos a serem reconhecidos através dos detectores de anomalias. Esses detectores monitoram as atividades da rede de computadores, correspondente ao organismo biológico.

2.3.5 Algoritmo das Células Dendríticas

Este algoritmo realiza a detecção de anomalias considerando o comportamento do ambiente de aplicação. Inicialmente, o algoritmo foi aplicado ao problema de detecção de intrusão e ao *SYN Scan* (GREENSMITH, 2007). Uma versão mais compacta foi definida em (GREENSMITH; AICKELIN, 2008). O algoritmo de células dendríticas é explicado da seguinte forma por Silva em (SILVA, 2009): em termos computacionais, a célula realiza uma fusão de dados multi-sensores baseando-se em janelas de tempo, o conjunto de células correlaciona os sinais e os antígenos, emitindo dados sobre o grau de anomalia de cada antígeno. As células dendríticas coletam algumas amostras de antígenos, armazenando-as. Então, estas são expostas aos sinais de entrada, que são processados e convertidos em sinais de saída usados na classificação do antígeno apresentado. O processo é repetido um determinado número de ciclos ou até que os antígenos tenham sido avaliados. Durante o processo, a atualização dos sinais de entrada, e o

cálculo dos sinais de saída, através da entrada, ocorrem na fase imatura da célula. Uma vez alcançado o valor de coestimulação, a célula realiza o processo de migração e, em seguida, reage de acordo com a concentração dos sinais coletados.

O funcionamento da inspiração biológica e do algoritmo poder ser resumido através da Figura 5:

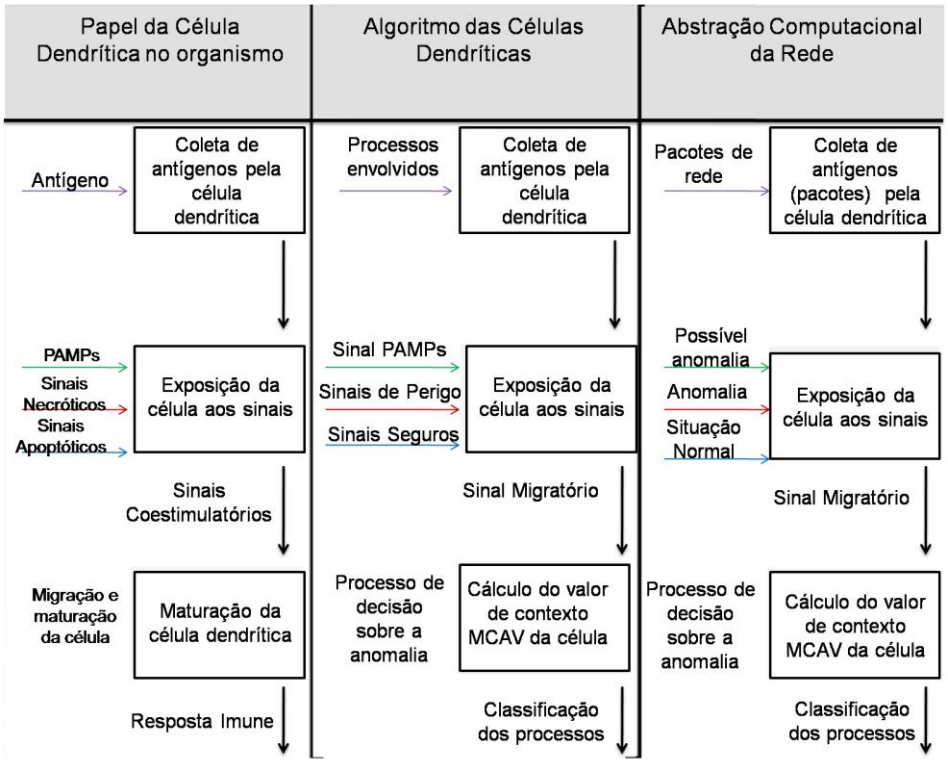

Figura 5 – Inspiração biológica e algoritmo das células dendríticas
Fonte: Adaptado de SILVA; ERRICO; CAMINHAS (2010)

- Padrão Molecular Associado ao Patogeno (*PAMPs*) - são sinais indicadores de uma situação anormal. Um aumento na concentração destes sinais conduz no aumento da coestimulação, que conduz à migração da célula e do sinal que torna a célula madura e classifica os antígenos como perigosos;
- Sinais de perigo - indicam a ocorrência de uma anormalidade, (situação de perigo), possuindo porém uma potência menos confidente que a do sinal PAMP e aumenta as concentrações dos sinais coestimulatórios e do sinal que torna a célula madura;

- Sinais seguros - este sinal é interpretado como uma ocorrência normal do sistema (situação de segurança) e, em uma concentração muito grande, aumenta o sinal de saída que torna a célula 'semi-madura' e reduz o valor cumulativo do sinal de saída que torna a célula madura;

Para o calculo do MCAV, faz-se o uso da Equação 1:

$$\mathrm{MCAV} = \frac{M}{\mathrm{Sm} + \mathrm{m}} \qquad (1)$$

MCAV é o índice ou probabilidade de anomalia de um antígeno. O MCAV é uma variável definida no intervalo entre 0 e 1, onde 0 indica uma situação possivelmente normal e 1 indica uma situação possivelmente anômala. Como o MCAV pode ser considerado uma variável de decisão, requer um limiar para a classificação de eventos. Onde Sm representa a quantidade de antígenos em células semi-maduras e M representa a quantidade de antígenos em células maduras.

Em (SILVA, 2009) são definidos as principais fases do algoritmo:

- Inicialização - é a fase que envolve a configuração de vários parâmetros do algoritmo. Após esta etapa, tem-se a fase de atualização;
- Atualização - é dividida em duas subetapas:
  - atualização de dados - processo contínuo onde as estruturas de dados são atualizadas em intervalos, os sinais são atualizados em intervalos de tempo e os antígenos em eventos;
  - Amostragem de dados - processo discreto onde os sinais e antígenos são acessados pelas células dendríticas. Inclui a atualização das células com novos valores dos sinais e dos antígenos e o processamento dos sinais de entrada, formando os sinais de saída;

- Agregação - todos os antígenos são analisados e o MCAV por antígeno é gerado.

Algoritmo genérico é definido pelo pseudocódigo na Figura 6, segundo (GREENSMITH,2007):

```
Algorithm 1: Pseudocode of the functioning of a generic DC object.
input : signals from all categories and antigen
output: antigen plus context values
initialiseDC;
while CSM output signal < migration Threshold do
  get antigen;
  store antigen;
  get signals;
  calculate interim output signals;
  update cumulative output signals;
end
cell location update to lymph node;
if semi-mature output > mature output then
  cell context is assigned as 0 ;
else
  cell context is assigned as 1;
end
kill cell;
replace cell in population;..
```

Figura 6 - Algoritmo genérico das células dendríticas.
Fonte: Greensmith (2007)

A base de dados utilizada para este algoritmo, onde se terá os dados necessários para as entradas será a DARPA KDD'99.

2.4 BASE DE DADOS DARPA KDDCUP'99

A base de dados escolhida para aplicação do algoritmo foi a DARPA KDDCUP'99. Ela é uma das poucas bases de dados de tráfego de rede disponíveis publicamente, devido a questões de legalidade, privacidade e segurança, como discutido em (PAXSON,2007). Apesar de ter sido criada a mais de dez anos, ela é a base mais utilizada para testar Sistemas de Detecção de Intrusão baseados em anomalia (TAVALLAEE et al.,2009).

A KDDCUP'99 foi concebida através da simulação de um ambiente de uma rede militar da força aérea dos Estados Unidos (*U.S. Air Force*). A rede foi operada em um ambiente real, alimentada por conexões *TCP dump*, mas sendo bombardeada por uma sequência de múltiplos ataques (RAMOS; SANTOS, 2011). É composta por cerca de 4 *GB* de arquivo comprimido de dados brutos de conexão *TCP*, referente a sete semanas de tráfego de rede, que podem ser transformado em cerca de 5 milhões de registros de conexões, cada um com 100 bytes. Duas semanas de testes tem aproximadamente dois milhões de registros de conexão (CAMPOS; LIMA,2012). Por ter um tamanho massivo, necessita-se a preparação da base de dados original, para que se possa gerar uma base de dados menor e suficiente para a aplicação do algoritmo. Para isso, utilizou-se o trabalho desenvolvido em (SOUZA et al., 2008).

O processo de preparação de dados foi realizado usando o SQL Server e a suite WEKA (UNIVERSITY OF WAIKATO, 2010). A massa de dados usada foi importada para o SQL Server e várias consultas foram feitas com o intuito de verificar se alguns atributos que tinham poucos valores distintos eram, de fato, atributos irrelevantes. No WEKA, foram calculadas as estatísticas dos dados, realizada a normalização e a seleção dos atributos (SOUZA et al., 2008).

Após as aplicações dos métodos definidos em Souza (SOUZA et al., 2008), obteve-se a base definida da forma que se pode observar na Quadro 2:

Atributo	**Descrição**	**Tipo**
duration	Tempo em segundos de conexão	*numeric*
protocol_type	Tipo de conexão	*icmp,tcp,udp*
service	Tipo de serviço no destino	*IRC, X11, Z39_50, aol, auth, bgp, courier, csnet_ns, ctf, daytime, discard, domain, domain_u, echo, eco_i, ecr_i, efs, exec, finger, ftp, ftp_data, gopher, harvest, hostnames, http, http_2784, http_443, http_8001, icmp, imap4, iso_tsap, klogin, kshell, ldap, link, login, mtp, name, netbios_dgm, netbios_ns, netbios_ssn, netstat, nnsp, nntp, ntp_u, other, pm_dump, pop_2,*

		pop_3, printer, private, red_i, remote_job, rje, shell, smtp, sql_net, ssh, sunrpc, supdup, systat, telnet, tftp_u, tim_i, time, urh_i, urp_i, uucp, uucp_path, vmnet, whois
flag	Estado da conexão (normal ou erro)	*OTH, REJ, RSTO, RSTOS0,RSTR, S0, S1, S2,S3, SF, SH*
land	1 se o host e a porta da origem e destino são os mesmos, 0 caso contrário	*0, 1*
wrong_fragment	Número de fragmentos “errados”	*numeric*
hot	Número de indicadores “importantes”	*numeric*
num_failed_logins	Número de tentativa de login com falha	*numeric*
logged_in	1 se o login obteve sucesso, e 0 caso contrário	*0,1*
num_compromised	Número de condições comprometedoras	*numeric*
root_shell	1 se o Shell root é obtido, 0 caso contrário	*numeric*
num_file_creations	Número de operações de criação de arquivos	*numeric*
num_access_files	Número de operações a arquivos de controle de acesso	*numeric*
count	Número de conexões para o mesmo host como conexão atual nos últimos 2 segundos	*numeric*
diff_srv_rate	% de conexões a diferentes Serviços	*numeric*
srv_diff_host_rate	% de conexões a diferentes hosts	*numeric*
dst_host_count	Atributo não documentado	*numeric*
dst_host_diff_srv_rate	Atributo não documentado	*numeric*
dst_host_srv_diff_host_rate	Atributo não documentado	*numeric*
dst_host_serror_rate	Atributo não documentado	*numeric*
dst_host_rerror_rate	Atributo não documentado	*numeric*
class	Classificação da conexão	neg_normal,pos_normal
src_bytes	Número de bytes da origem ao destino	*numeric*
dst_bytes	Número de bytes do destino à origem	*numeric*

Quadro 2 – Atributos básicos de uma conexão TCP.
Fonte:(SOUZA et al., 2008).

3 METODOLOGIA

A partir do estudo de sistemas imunes para sistemas de detecção de intrusos e o modelo do perigo, implementou-se o algoritmo das Células Dendríticas. A linguagem utilizada foi o C, devido ao seu uso em sistemas de detecção de intrusos *open-source*, o que facilitará a incorporação do algoritmo nestes sistemas em trabalhos futuros.

Os dados da base, utilizados, foram obtidos por meio da página oficial da edição de 1999 do DARPA KDDCUP'99 (KDD Cup 1999 Data, 1999), por ela ser uma das poucas bases de dados reais disponíveis, devido a fatores legais. Por meio desta base de dados foi efetuada a extração dos dados, nesta fase foram aplicados todos os passos de pré-processamento descrito em Greensmith (Greensmith, 2007).

Após o estudo de vários algoritmos utilizados em *IDS* (*Intrusion Detection System*), foram selecionados os melhores algoritmos de acordo com suas performances. Estas performances foram extraidas dos métodos apresentados no trabalho de Garg (GARG; KHURANA, 2014), onde se tem a comparação de vários algoritmos em que os mesmos são elencados do melhor ao pior.

Com os algoritmos selecionados e o algoritmo das células dendríticas implementado, foram efetuados os testes, onde foram submetidos à mesma base de dados. Nos algoritmos selecionados para fazer a comparação, utilizou-se a suite WEKA.

Com os resultados concluídos, para a comparação e validação dos dados foi utilizado a matriz de confusão. Com esta métrica foram verificados os resultados e feitas as conclusões, onde a partir disto pode-se verificar os possíveis trabalhos futuros.

3.1 FERRAMENTA DE SOFTWARE UTILIZADO

A suite WEKA é formada por um conjunto de implementações de algoritmos de diversas técnicas de Mineração de Dados (UNIVERSITY OF WAIKATO, 2010).

Este software foi desenvolvido na linguagem Java sob domínio da licença GPL[1], com o principal objetivo a portabilidade, podendo executá-lo nos mais diversos sistemas operacionais assim como utilizar-se dos benefícios da orientação a objetos.

Alguns métodos implementados no WEKA são:

- Métodos de classificação:
  - Árvore de decisão induzida;
  - Regras de aprendizagem;
  - *Naive Bayes;*
  - Tabelas de decisão;
  - Regressão local de pesos;
  - Aprendizado baseado em instância;
  - Regressão lógica;
  - *Perceptron;*
  - *Perceptron* multicamada;
  - Comitê de *perceptrons;*
  - *SVM.*
- Métodos para predição numérica:
  - Regressão linear;
  - Geradores de árvores modelo;
  - Regressão local de pesos;
  - Aprendizado baseado em instância;
  - Tabela de decisão;
  - *Perceptron* multicamadas.
- Métodos de Agrupamento:
  - EM;
  - *Cobweb*;
  - *SimpleKMeans*;
  - *DBScan*;
  - CLOPE.

[1] *General Public License*, licença para software livre.

- Métodos de Associação:
  - Apriori;
  - FPGrowth;
  - *PredictiveApriori;*
  - Tertius.

Este software utilizará arquivos do tipo ARFF. As especificações deste arquivo serão introduzidas a seguir.

3.2 ARQUIVO ARFF

Para a utilização de técnicas de mineração de dados precisa-se que os dados estejam de forma organizada. O WEKA possui um formato para a organização dos dados denominado ARFF. Nele deve conter algumas informações, como o domínio do atributo (valores que os atributos podem representar e atributo classe).

Primeiramente neste arquivo define-se o tipo do atributo e/ou os valores representados, sendo que estes valores devem estar entre chaves ({}) separados por vírgulas.

Em seguida o arquivo ARFF é composto pelas instâncias presentes nos dados, os atributos de cada instância devem ser separados por vírgula, e aqueles que não contêm valor, devem ser representados pelo caractere '?'.

As informações presentes no arquivo ARFF são especificadas utilizando marcações. Por exemplo, o nome do conjunto de dados é especificado através da marcação @*relation*, @*atribute* para os atributos, e os dados em si são definidos através da marcação @*data*.

Exemplo de um arquivo pode ser visto na Figura 7.

```
@relation kddcup99

@attribute count numeric
@attribute dst_host_count numeric
@attribute dst_host_rerror_rate numeric
@attribute src_bytes numeric
@attribute dst_bytes numeric
@attribute class {neg_normal,pos_normal}

@data
0.001957,1,0,0.000002,0.000006,pos_normal
0.001957,1,0,0.000002,0.000006,pos_normal
0.001957,1,0,0.000002,0.000006,pos_normal
0.003914,1,0,0.000002,0.000006,neg_normal
0.003914,1,0,0.000002,0.000006,neg_normal
0.003914,1,0,0.000002,0.000006,neg_normal
0.003914,0.039216,0,0,0,pos_normal
0.001957,1,0,0.000002,0.000006,pos_normal
0.003914,1,0,0.000002,0.000006,neg_normal
```

Figura 7 - Exemplo de formato de arquivo ARFF
Fonte: Autoria Própria

4 EXPERIMENTOS E RESULTADOS

Os experimentos foram efetuados em um computador com o sistema operacional Linux Debian 6, 4 GB de memória RAM e processador Intel Core I5 com 2.5 GHz.

A aplicação para a validação do algoritmo das células dendríticas foram efetuados conforme os passos descritos em (Greensmith, 2007). Como citado anteriormente a base de dados utilizada foi a KDDCUP`99 visando resultados mais próximos da realidade. E, ao final, validada por meio de comparações que são descritas nesta seção, por meio do software denominado WEKA.

4.1 PRÉ-PROCESSAMENTO

A preparação desta base de dados para aprendizado de máquina consistiu em seguir algumas etapas. Durante estas etapas os atributos foram normalizados e selecionados para dar origem aos sinais de *PAMP*, seguro e de perigo. Além desses processos foi necessário estabelecer um limiar de anomalia de acordo com os dados das classes anomalia e normal.

Todos esses processos serão exemplificados e explicados na sequência com valores reais dos experimentos.

A DARPA KDDCUP`99 é uma base de dados extensa, provida de eventos reais que foram capturados em uma rede real. Devido ao seu tamanho esta base requer um grande processamento, em (SOUZA A.; SILVA G., 2008) os autores reduzem a base em 24 atributos levando em conta os atributos mais relevantes para ataques em uma rede, sendo estes já descritos anteriormente.

De acordo com o pré-processamento, descrito em (GREENSMITH 2007), a base de dados passou de 24 para 5 atributos. No pré-processamento de todos os 24 atributos foram extraídos os do tipo contínuo, os categóricos não foram utilizados. Foram extraídos a média, o desvio padrão e a mediana de cada uma das variáveis apresentadas. De todos os atributos foram selecionados cinco, a seleção ocorreu

por meio do desvio padrão em que foram escolhidos os atributos com o desvio padrão mais alto. No anexo A apresenta-se o quadro com os respectivos desvios padrões. Os atributos selecionados foram:

- *count* - número de conexões para o mesmo host como conexão atual nos últimos 2 segundos *;*
- *dst_host_count* – número de conexões dos últimos 2 segundos que tenham o mesmo host destino*;*
- *dst_host_rerror_rate* – número de conexões que foram rejeitadas e possuam o mesmo host destino*;*
- *src_bytes* - número de bytes da origem ao destino;
- *dst_bytes* - número de bytes do destino à origem;

4.1.1 Sinal Seguro e PAMP

Para o cálculo do sinal seguro e PAMP, foi utilizado o atributo com menor desvio padrão dentre os cinco.

O cálculo destes sinais são efetuados da seguinte forma:

- Seleção do atributo com o desvio padrão mais baixo:
  - Neste caso o atributo selecionado é *dst_host_count* com o desvio padrão de 0,239.
- Calcula-se a mediana do atributo:
  - Mediana de *dst_host_count* é igual a 1.
- Para cada valor do atributo calcula-se o sinal de PAMP e seguro como no algoritmo ilustrado na Figura 8:

```
If valor > mediana then
        valor é um sinal seguro;
        sinal seguro = |média – valor do atributo|;
        sinal PAMP = 0;

else
    valor é um sinal PAMP;
    sinal PAMP= |média – valor do atributo |;
    sinal seguro= 0;
end
```

Figura 8 - Algoritmo : Processo para calculo dos sinais de PAMP e seguro.
Fonte: Autoria Própria

Por exemplo, se o valor for 8 e a mediana igual a 1 o sinal será seguro, logo, o sinal recebe a média -8 em módulo e o sinal de PAMP recebe zero. Se a o valor for 0.5 e a mediana 1 o sinal será de PAMP, então sinal PAMP receberá a média -0.5 em módulo e o sinal seguro será igual a zero.

4.1.2 Sinal de Perigo

Para o processo de obtenção do sinal de perigo foram utilizadas às seguintes regras:

1) Seleção dos 4 atributos com o desvio padrão mais alto:
    - *count;*
    - *dst_host_rerror_rate;*
    - *src_bytes ;*
    - *dst_bytes;*

2) Para cada atributo, calcula-se a média dos elementos da classe normal, apresentado na Tabela 1.

Tabela 1- Média dos elementos da classe normal.

Atributo	Média
count	0.527
dst_host_rerror_rate	0.143
src_bytes	1.288
dst_bytes	0.129

Fonte: Autoria Própria

3) Para cada atributo, calcula-se a distância absoluta entre os atributos e as médias como na tabela 4:

Tabela 2 – Média e distância absoluta entre os atributos

Atributo	Valor	Média	Distância absoluta
count	0.001957	0.527	0.525043
dst_host_rerror_rate	0	0.143	0.143
src_bytes	0.078067	1.288	1.209933
dst_bytes	0.017561	0.129	0.111439

Fonte: Autoria Própria

4) Divide-se os valores das distâncias absolutas pelo número de atributos, como pode ser observados na Equação 2.

$$SP = \frac{\sum(\text{ distâncias absolutas})}{\text{numero de atributos}} \quad (2)$$

$$SP = \frac{(0.525043 + 0.143 + 1.209933 + 0.111439)}{4}$$

$$SP = 0.49735375$$

5) O resultado é o sinal de perigo.

Para os dados acima o sinal de perigo é 0.49735375.

Neste processo também é calculado um limiar de anomalia, sendo este obtido da seguinte maneira:

- Efetua-se a divisão do numero de anomalias pelo numero total de instâncias, como na Equação 3.

$$La = \frac{AN}{\text{Total}} \quad (3)$$

$$La = \frac{250436}{311029}$$

$$La = 0{,}80518536856$$

4.1.3 Antígeno

O antígeno usa o índice, na ordem em que eles são dados. Por exemplo, a primeira linha é igual a 1, segunda linha igual 2 e assim continuamente.

4.1.4 Normalização dos Sinais

Ao final é necessário normalizar os sinais assim como descrito em (Silva; Palhares; Caminhas, 2012), segundo o mesmo, a etapa de normalização dos dados é um fator de grande importância, sobretudo na geração dos sinais de entrada do DCA, que seguem determinados valores. A normalização deriva das seguintes formas:

- Adotar faixas de valores para os mesmos, conforme um exemplo na Equação 4, válido para os sinais PAMP e perigo. Onde DSmin e DSmax são limiares para o sinal, Vmaxd é o valor máximo estipulado e k é o instante do sinal.

$$DS(k) = \begin{cases} 0, & DS(k) < DSmin \\ \frac{DS(k) - DSmin}{DSmax - DSmin}, & DSmin < DS(k) < DSmax \\ \text{Vmaxd}, & DS(k) > DSmax \end{cases} \quad (4)$$

- Para o sinal seguro, o processo é diferente, pois o valor deve ser invertido para sua utilização correta, conforme o exemplo na Equação 5. Vale ressaltar que o valor de Vmaxs e Vmaxd podem ser diferentes.

$$SS(k) = \begin{cases} \text{Vmaxs}, & SS(k) < SSmin \\ \frac{DSmax - SS(k)}{SSmax - SSmin}, & SSmin < SS(k) < SSmax \\ 0, & SS(k) > SSmax \end{cases} \quad (5)$$

Após o pré-processamento dos dados, os sinais foram enviados ao algoritmo para os antígenos serem processados.

Na figura 9 pode-se observar um arquivo de entradas válido.

```
1 signal 0.108797 0.277138
2 signal 0.108797 0.277138
3 signal 0.108797 0.277138
4 antigen 4
5 antigen 5
6 antigen 6
7 signal 0.851987 0.295166
8 signal 0.108797 0.277138
9 antigen 9
10 signal 0.612772 0.252565
11 antigen 11
12 signal 0.879438 0.250476
13 signal 0.108797 0.277138
14 antigen 14
15 signal 0.679438 0.667688
16 signal 0.197085 0.299344
```

Figura 9- Exemplo de formato de entrada válido
Fonte: Autoria Própria

4.2 RESULTADOS

Para a análise da eficiência do algoritmo desenvolvido, mediante a base de dados selecionada, utilizou-se o trabalho de (GARG; KHURANA, 2014) no qual houve o estudo de técnicas de classificações. Basicamente existem oito categorias de classificadores e cada categoria contém diferentes algoritmos de aprendizagem de máquina (GARG; KHURANA, 2014).

Neste estudo foram testados quarenta e cinco classificadores, sendo estes ordenados do melhor para o pior.

Para a comparação com o algoritmo das células dendríticas foram selecionados os onze melhores, aproximadamente 25% do total dos algoritmos, sendo eles:

- *Rotation Forest* - neste método o conjunto de recursos está dividido aleatoriamente em subconjuntos K (K é um parâmetro do algoritmo) e a análise de componentes principais é aplicada a cada subconjunto. Para isto utiliza-se a abordagem rotação para incentivar precisão simultaneamente individual e diversidade dentro do conjunto. Diversidade é promovida através da extração de características. Usa-se neste método as árvores de decisão, porque elas são sensíveis a rotação dos eixos de recursos, daí o nome "floresta";
- *Random Tree* - este classificador é uma árvore de decisão que considera apenas alguns atributos, escolhidos aleatoriamente para cada nó da árvore, onde são utilizados na classificação de novos objetos;
- *Random Committee* - utiliza classificadores que tem funcionamento aleatório como base. Cada modelo de classificação gerado é construído usando uma semente de número aleatório diferente (mas baseado nos mesmos dados). A previsão final é uma média das previsões geradas pelos modelos base individuais;
- *Random Forest* - este algoritmo de comitê é um conjunto de árvores de classificação. Cada árvore dá um voto que indica sua decisão sobre a classe do objeto. A classe com o maior número de votos é escolhida para o objeto*;*
- *IBK* - o algoritmo IBK e um algoritmo de aprendizagem baseado em instâncias. Esse tipo de algoritmo é derivado do método de classificação de k-vizinhos mais próximos. Nele existe uma função de similaridade que obtém um valor numérico pelo cálculo da distância euclidiana. Então a classificação gerada para um padrão i será influenciada pelo resultado da classificação dos seus k-vizinhos mais próximos;
- *Random Sub Space* - é uma técnica de combinação de modelos onde os dados de treinamento são amostrados no espaço de características, sendo os classificadores treinados sobre subespaços, aleatoriamente escolhidos do espaço original de treinamento;

- *IB1* - usa a distância euclidiana normalizada para encontrar a instância de treinamento mais próximo do exemplo dos dados de teste, logo prevê a mesma classe que está instanciada no treinamento. Se várias instâncias têm a mesma distância para a instância de teste, o primeiro encontrado será usado;
- *Part* - baseado em regras de decisões, e utiliza internamente o algoritmo C4.4. Ele constrói árvores de decisão parciais a cada iteração e transforma a melhor folha da árvore atual em uma regra. Após escolher a melhor forma, o algoritmo retira todas as instâncias que se encaixem na regra gerada pela folha, para gerar uma nova árvore, e, por conseguinte, uma nova regra*;*
- *Jrip* - método baseado em regras, utiliza o algoritmo IREP para obter um conjunto de regras, dando origem a um modelo inicial. Este modelo é simplificado de forma iterativa, através da poda incremental repetida, para a redução do erro*;*
- *NB Tree* - utiliza um modelo híbrido, uma combinação de árvores de decisão com *naive-Bayes*. Neste modelo os nodos contêm divisões considerando um único atributo, como nas árvores de decisão regulares, porém os nodos das folhas possuem classificadores de *naive-Bayes*;
- *J48* - usa o método divisão e conquista para aumentar a capacidade de predição das árvores de decisão. Assim, sempre usa o melhor passo avaliado localmente, sem se preocupar se esse passo vai produzir a melhor solução, cada problema é dividido em vários subproblemas, sendo criadas sub-árvores entre a raiz e as folhas.

Esses algoritmos foram submetidos à mesma base de dados utilizada no algoritmo das células dendríticas, por meio da suíte WEKA. A partir dos resultados foram levantados os dados para a comparação, sendo eles:

- Classificação Correta: Quantidade de vezes que o algoritmo acertou a resposta;
- Classificação incorreta: Quantidade de vezes que o algoritmo errou a resposta;

- Falso positivo (FP): Ação é classificada como uma possível intrusão, mas se trata de uma ação normal;
- Falsos negativos (FN): Quando ocorre uma ação intrusiva, porém ela é classificada como uma ação normal;
- Verdadeiros positivos (VP): Ação é classificada como uma possível intrusão, e realmente se trata de uma intrusão;
- Verdadeiros Negativos (VN): Quando ocorre uma ação intrusiva e ela é classificada como uma intrusão.

Os resultados comparativos dos algoritmos podem ser vistos na Tabela 3.

Tabela 3 – Resultados comparativo dos algoritmos

Algoritmo	Classificação Correta	Classificação Incorreta	Falsos Positivos	Falsos Negativos	Verdadeiros Positivos	Verdadeiros Negativos
DCA	99,460500%	0,539499%	0	1678	65530	248728
Random Tree	97,687400%	2,312600%	3663	3530	56930	246906
Random Committee	97,687400%	2,312600%	3663	3530	56930	246906
IBK	97,687400%	2,312600%	3663	3530	56930	246906
Random Forest	97,675800%	2,324200%	3638	3591	56955	246845
Rotation Forest	97,517600%	2,482400%	3827	3894	56766	246542
Jrip	97,040800%	2,959200%	5386	3818	55207	246618
Random Sub Space	97,003200%	2,996800%	1060	8261	59533	242175
NB Tree	96,934400%	3,065600%	4714	4821	55879	245615
IB1	95,193400%	4,806600%	4683	10267	55910	240169
J48	94,616300%	5,383700%	2271	14474	58322	235962
Part	94,527500%	5,472500%	2138	14883	58455	235553

Fonte: Autoria Própria

Pode-se notar que o algoritmo das células dendríticas teve um resultado muito melhor que os demais, tendo uma classificação correta de mais de 99% da base de dados.

Houve uma redução nos falsos positivos devido a forma como que o algoritmo funciona, esta redução já havia sido notada em (GREENSMITH, 2007) sendo este fator de grande valia para um sistema de detecção de intrusos.

Na análise desses dados a mais significativa é a visivel redução de falsos negativos, a qual tem-se uma intrusão. Porém ela é classificada como uma ação normal. Este fator é o principal motivo que gera prejuízo a empresas devido ao

roubo de informações e violações de dados. Verificando que esta redução caiu para aproximadamente 0,5% será de grande valía para a detecção dessas ameaças.

Com a redução dos falsos positivos e falsos negativos pode-se notar que o DCA pode ser um possível algoritmo a ser implementado em um sistema de detecção de intrusos.

5 CONCLUSÕES

Esta seção apresenta as considerações finais sobre o estudo, confrontando os objetivos gerais e específicos com os resultados obtidos através dos experimentos realizados.

O objetivo geral de implementar um algoritmo baseado em sistemas imunes artificiais através do modelo do perigo para um sistema de detecção de intrusos, tendo por objetivo a análise e comparação do mesmo com os métodos da literatura assim como os utilizados atualmente foi alcançado por meio da metodologia descrita.

A metodologia engloba as seguintes fases: pré-processamento da base de dados, criação dos sinais de estímulos, aplicação do algoritmo e avaliação dos dados comparativos de determinados algoritmos fornecidos pelo WEKA.

Quanto ao objetivo específico de estudar a base de dados KDD Cup 99, implementar o algoritmo das células dendríticas e a utilização do WEKA, o referencial teórico levantado apresenta a validação para o mesmo.

Em relação ao objetivo especifico de realizar os experimentos comparativos com os algoritmos classificadores utilizados na atualidade e na literatura foi alcançado, pois os resultados sugerem que este algoritmo teve um resultado muito superior aos demais, reduzindo os falsos positivos e falsos negativos utilizando uma base de dados mais próxima a realidade.

Assim é possível concluir que esse trabalho cumpriu com todos os objetivos propostos.

5.1 TRABALHOS FUTUROS

O algoritmo das células dendríticas teve um desempenho superior aos demais algoritmos, e também apresentou o mesmo padrão de redução de falsos positivos e falsos negativos vistos em outros trabalhos como em (GREENSMITH; AICKELIN, 2008) e (SILVA, 2009).

Com base nesses resultados, como sugestão para um trabalho futuro pode-se citar a implementação do algoritmo das células dendríticas em um sistema de detecção de intruso para a verificação de como este algoritmo se comporta com situações em tempo real, para comprovar se o processamento dos sinais é realmente feitos em tempo hábil e com as mesmas taxas de acertos e erros assim como os falsos positivos, verdadeiros positivos, falsos negativos e verdadeiros negativos.

REFERÊNCIAS

ABNT. **Tecnologia da informação – Técnicas de segurança – Código de Prática para a gestão da segurança da informação** – NBR ISO/IEC 27002. *Tecnologia da informação – Técnicas de segurança – Código de Prática para a gestão da segurança da informação – NBR ISO/IEC 27002* .2005

ALMEIDA, C.; PALHARES, R.; CAMINHAS, W. **Design of an artificial immune system based on danger model for fault detection**. *Expert Systems with Applications* , 5145–5152, 2010.

CERT.br. *CERT.br* - **Centro de Estudos, Respostas e Tratamento de Incidentes de Segurança no Brasil**. Disponível em: <http://www.cert.br/stats/incidentes/2014-jan-dec/tipos-ataque-acumulado.html>. Acesso em: 12 set. 2015.

CORMACK, D. ***HAM Histologia****.* (R. d. Janeiro, Ed.) Rio de Janeiro, 1991.

DEBAR, H.; DACIER, M.; WESPI, A. ***A revised taxonomy for intrusion-detection systems*** (Vol. 55), 2000.

FEILY, M.; SHAHRESTANI, A.; RAMADASS, S. **A survey of botnet and botnet detection**. *IEEE*, (pp. 268-273), 2009.

FIGUEREDO, G. P.; BERNARDINO, H. S.; CORRÊA, H. J. ***Introdução aos Sistemas Imunológicos Artificiais****,* 2013*.*

GARCIA-TEODORO, P. **Anomaly-based network intrusion detection: Techniques, systems and challenges**. *computers \& security , 28* (1), 18-28, 2009.

GARG, T.; KHURANA, S. **Comparison of Classification Techniques for Intrusion Detection Dataset Using WEKA**. *IEEE International Conference on Recent Advances and Innovations in Engineering* , 2014.

GREENSMITH, J. ***The Dendritic Cell Algorithm****.* Ph.D. dissertation, University of Nottingham, 2007.

GREENSMITH, J.; AICKELIN, U. **The deterministic dendritic cell algorithm**. In *Artificial Immune Systems* (pp. 291-302). Springer, 2008.

GUANGMIN, L. (2008). **Modeling Unknown Web Attacks in Network Anomaly Detection**. *Third International Conference on Convergence and Hybrid Information Technology* , 112-116.

KDD CUP 1999 DATA. ***KDD Cup 1999 Data*** . Disponível em: <http://kdd.ics.uci.edu/databases-/kddcup99/kddcup99.html>. Acesso em: 05 fev. 2014.

LI, L., & LEE, G. **DDoS attack detection and wavelets**. *12th International Conference on Computer Communications*, 2003.

LIMA, L. M., & LIMA, A. S. Sistema para Detecção de Intrusão em Redes de Computadores com Uso de Técnica de Mineração de Dados, 2012.

LIU, T. E. **Method for network anomaly detection based on Bayesian statistical model with time slicing**. *7th World Congress on Intelligent Control and Automation.* , 3359-3362, 2008.

MATZINGER, P. **Tolerance, danger, and the extended family**. *Annual review of immunology , 12* (1), 991-1045, 1994.

PAXSON, V. **Considerations and pitfalls for conducting intrusion detection research**. *International Computer Science Institute and Lawrence Berkeley National Laboratory Berkeley, California USA*, 2007.

PERLIN, T.; NUNES, R.; KOZAKEVICIUS, A. **Detecção de Anomalias em Redes de Computadores através de TransformadasWavelet.** *Revista Brasileira de Computação Aplicada , 3* (1), p. 02-15, 2011.

PONEMON INSTITUTE; IBM. *Estudo do Custo de Violações de Dados 2014: Análise Global,* 2014.

RAMOS, A. L.; DOS, C. N. **Combinando Algoritmos de Classificação para Detecção de Intrusão em Redes de Computadores**. *Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, 2011.

SABAHI, F.; MOVAGHAR, A. **Intrusion detection**: A survey. *IEEE*, (pp. 23-26), 2008.

SAMAAN, N.; KARMOUCH. A. **Network anomaly diagnosis via statistical analysis and evidential reasoning**. *IEEE Transactions on, v. 5, n. 2* , 65-77, 2008.

SELVAKANI, S.; RAJESH, R. S. **Genetic Algorithm for framing rules for intrusion Detection**. *International Journal of Computer Science and Network Security*, 2007.

SILVA, G. ***Detecção de Intrusão em Redes de Computadores: Algoritmo Imunoinspirado Baseado na Teoria do Perigo e Células Dendríticas***. Universidade Federal de Minas Gerais, 2009.

SILVA, G.; PALHARES, R.; CAMINHAS, W. **Introdução Ao Algoritmo Das Células Dendríticas No Contexto De Detecção De Falhas Em Sistemas Dinâmicos**. *Congresso Brasileiro de Automática*, 2012.

SOUZA, A. J.; SILVA, A. G. **Mineração de Dados para Detecção de Intrusão de Redes**. Universidade Federal de Pernambuco - UFPE , 2008.

STALLINGS, W. ***Cryptography and Network Security****, 4/E.* Pearson Education India, 2006.

SUNDARAM, A. ***An introduction to intrusion detection****,* 2009*.*
Disponível em: < http://www.acm.org/crossroads/xrds2->. Acesso em: 08 out. 2014.

SZYMCZYK, M. **Detecting botnets in computer networks using multi-agent technology.** *IEEE*, (pp. 192-201), 2009.

TAVALLAEE, M.; BAGHERI, E.; LU, W.; GHORBANI, A. A. **A Detailed Analysis of the KDD CUP 99 Data Set**. *In Proceedings of the Second IEEE Symposium on Computational Intelligence in Security and Defense Applications*, 2009.

TJADEN, B. C. ***Computer, Internet and Network Systems Security****.* A B F Content, 2001.

UNIVERSITY OF WAIKATO. (2010). *WEKA 3 – **Machine Learning Software in Java***. University of Waikato: WEKA software.
Disponível em: < http://www.cs.waikato.ac.nz/ml/WEKA/>. Acesso em: 20 mai. 2015.

YAO, L.; ZHITANG, L.; SHUYU, L. **A Fuzzy Anomaly Detection Algorithm for IPv6**. In: Second International. 67, 2006.

ZHU, Z.; LU, G.; CHEN, Y.; FU, Z., ROBERTS, P.; HAN, K. ***Botnet Research Survey***., (pp. 967-972), 2008.

ANEXO A – SELEÇÃO DE ATRIBUTOS A PARTIR DO DESVIO PADRÃO

Seleção de Atributos a Partir Do Desvio Padrão

Nome	Tipo	Desvio Padrão
duration	Contínuo	0,007
protocol_type	Categórico	Não aplicavél
Service	Categórico	Não aplicavél
flag	Categórico	Não aplicavél
land	Categórico	Não aplicavél
wrong_fragment	Contínuo	0,013
hot	Contínuo	0,003
num_failed_logins	Contínuo	0,01
logged_in	Categórico	Não aplicavél
num_compromised	Contínuo	0,002
root_shell	Categórico	Não aplicavél
num_file_creations	Contínuo	0,002
num_access_files	Contínuo	0,007
count	Contínuo	0,43
diff_srv_rate	Contínuo	0,107
srv_diff_host_rate	Contínuo	0,125
dst_host_count	Contínuo	0,239
dst_host_diff_srv_rate	Contínuo	0,096
dst_host_srv_diff_host_rate	Contínuo	0,036
dst_host_serror_rate	Contínuo	0,231
dst_host_rerror_rate	Contínuo	0,344
class	Categórico	Não aplicavél
src_bytes	Contínuo	94,912
dst_bytes	Contínuo	1,939

Fonte: Adaptado de Souza A (2008)