ULBRA CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS
COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

Luane Gomes Cunha

IMPLEMENTAÇÃO DO ALGORITMO DE SELEÇÃO NEGATIVA PARA DETECÇÃO DE *SPAMS*

Palmas

2012

Luane Gomes Cunha

IMPLEMENTAÇÃO DO ALGORITMO DE SELEÇÃO NEGATIVA PARA DETECÇÃO DE *SPAMS*

Trabalho apresentado como requisito parcial da disciplina de Trabalho de Conclusão de Curso I e II (TCC I e TCC II) do curso de Sistemas de Informação, orientado pelo Professor Mestre Fernando Luiz de Oliveira.

Palmas

2012

Luane Gomes Cunha

IMPLEMENTAÇÃO DO ALGORITMO DE SELEÇÃO NEGATIVA PARA DETECÇÃO DE *SPAMS*

Trabalho apresentado como requisito parcial da disciplina de Trabalho de Conclusão de Curso I e II (TCC I e TCC II) do curso de Sistemas de Informação, orientado pelo Professor Mestre Fernando Luiz de Oliveira.

Aprovada em 21 de junho de 2012.

BANCA EXAMINADORA

_____

Prof. M.Sc. Fernando Luiz de Oliveira.

Centro Universitário Luterano de Palmas

_____

Prof. M.Sc. Fabiano Fagundes

Centro Universitário Luterano de Palmas

_____

Prof. M.Sc. Madianita Bogo

Centro Universitário Luterano de Palmas

Palmas

2012

RESUMO

O Sistema Imunológico Artificial, baseado no Sistema Imune Humano, possui características que são interessantes para serem aplicadas na área da computação como, por exemplo, a aplicação desta técnica para a proteção de computadores. Uma vez que a segurança é um dos itens mais importantes, este trabalho tem como proposta o desenvolvimento de uma aplicação baseada no algoritmo de Seleção Negativa para a detecção de *spams*. *Spam* é uma ameaça que pode comprometer a integridade dos dados uma vez que podem disseminar vírus, além de comprometer a credibilidade de um usuário ou servidor de e-mail. O desenvolvimento da aplicação ocorrerá em três fases: a primeira consiste na geração de detectores; a segunda na avaliação destes; e a terceira fase incide no monitoramento, no qual o algoritmo será capaz de detectar um *spam*.

PALAVRAS-CHAVE: Sistema Imunológico Artificial, *Spams*, Seleção Negativa

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

aiNet - *Artificial Immune Network*

ASN – Algoritmo de Seleção Negativa

API - *Application programming interface*

BCR - B *cell receptor*

CLONALG - C*lonal selection algorithm*

CSA - *Clonal Selection Algorithm*

DCA - Algoritmo das Células Dendríticas

DNSBL - *Domain Name System Blacklist;*

HTTP - *HyperText Transfer Protocol*

IDE - *Integrated Development Environment*

IMAP - *Internet Message Access Protocol*

MHC - *Major Histocompatibility Complex*

NPC - *Non-Player Character*

OCR - *Optical Character Recognition*

POP3 - *Post Office Protocol*

RNA - Redes Neurais Artificiais

RNSA - *Real-Valued Negative Selection Algorithm*

SIA - Sistema Imunológico Artificial

SMS - *Short Message Service*

SMTP - *Simple Mail Transfer Protocol*

TCR - T *cell receptor*

Th - T *helper*

SUMÁRIO

1 INTRODUÇÃO

O homem é um dos seres mais complexos devido a sua evolução, capacidade de comunicação e de ser capaz de viver em sociedade. Existem vários estudos a fim de entender o funcionamento do corpo humano, desde estudos relacionados ao DNA até a psicologia. Dentre esses estudos, há um ramo da biologia que estuda o corpo humano e seus sistemas, sendo um deles o Sistema Imunológico, também chamado de Sistema Imune, que é o responsável pela defesa do organismo. Segundo Ferron e Rancano (2007, p. 57), o Sistema Imunológico "constitui o principal elemento defensivo do corpo humano e é formado por uma complexa estrutura interna que se encarrega de defender o interior do organismo [...]". Tais estruturas internas possuem características interessantes, como: adaptação, evolução, memória etc., que também são desejáveis na área da computação.

Há muito tempo vem sendo estudado e relacionado conhecimento de outras áreas, como a biologia, com a área de computação e informática, a fim de se obter resultados similares, ou seja, tenta-se reproduzir de maneira artificial as características da área estudada para que se possam obter soluções computacionais melhores. Existem vários estudos que validam essas pesquisas, como Redes Neurais Artificiais (RNA), técnica computacional que explora a inteligência artificial tendo como base o sistema nervoso; Algoritmos Genéticos, técnica que explora a evolução das espécies para encontrar soluções melhores para problemas de otimização e busca; e Agentes Inteligentes, que é uma técnica de inteligência artificial baseada em agentes do mundo real, no qual o agente computacional tem a capacidade de interagir com o ambiente, agregar informações, etc., sendo que isso possibilita uma melhor tomada de decisão no ambiente ao qual ele está inserido.

A técnica baseada no Sistema Imunológico Artificial (SIA) tem como objetivo simular as características do sistema inume natural na área da computação ou afins. No caso deste trabalho, o objetivo é estudar esta referida técnica e desenvolver uma aplicação baseada em um dos modelos do SIA, mais especificamente no modelo de Seleção Negativa, para realizar a detecção de mensagens de e-mails como sendo *spams* ou não.

Dessa forma, são apresentadas nas próximas seções o Referencial Teórico (seção 2), que abordará conceitos sobre e-mails, *spams* e técnicas de detecção, e também acerca do Sistema Imunológico Natural, suas principais funções e características; O Sistema Imunológico Artificial, suas características, aplicações e ferramentas e, ao final da seção,

serão apresentados alguns trabalhos relacionados ao tema estudado. Na seção de Materiais e Métodos (seção 3) há a apresentação dos tipos de materiais utilizados para o desenvolvimento do trabalho bem como da metodologia adotada para o seu desenvolvimento. Na seção de resultados e discussão (seção 4) são apresentados os resultados obtidos. As Considerações Finais (seção 5) contém a importância da pesquisa a cerca do tema abordado assim como trabalhos futuros. Por fim, as Referências Bibliográficas (seção 6) utilizadas na realização do trabalho.

2 REFERENCIAL TEÓRICO

Esta seção tem por objetivo apresentar os conceitos importantes para o entendimento do trabalho. Para tanto, são abordados conceitos relacionados a e-mail e *spam*, como definição, tipos de *spams*, técnicas de detecção e algumas ferramentas. Logo em seguida, as funções e características que abrangem o Sistema Imunológico Natural, bem como o Sistema Imunológico Artificial, seus conceitos, ferramentas e aplicações.

2.1. E-mail

O e-mail é um meio de comunicação assíncrona, ou seja, não requer que o destinatário esteja conectado para receber a mensagem. Assim como o correio da vida real, o correio eletrônico é destinado para a troca de mensagens, porém essas mensagens são eletrônicas e possuem características bem elaboradas, como, anexos, hiperlinks, textos em formato HTML e fotos (ROSS e KUROSE, 2010, p. 87).

O correio eletrônico, da forma que conhecemos hoje, nasceu em 1971 com a inserção do termo "@" separando o nome de usuário do nome da máquina. Inicialmente, as mensagens de texto eram simples e tinham tamanho limitado. Porém, com o passar do tempo este recurso foi ganhando novas funcionalidades, como o envio de arquivos em anexo e textos em formatos HTML, além do aumento da capacidade de transferência e de armazenamento. O formato do e-mail foi definido pelo RFC[1] (*Request For Comments*) 822 de 1982 que especifica os campos do e-mail (RFC 822, 1982, online). Além de possuir um padrão, o e-mail também possui uma série de protocolos e agentes que operam em seu funcionamento. Essa estrutura do e-mail será apresentada na seção a seguir.

2.1.1. Estrutura do e-mail

Com uma estrutura bem definida, o e-mail trabalha com protocolos e agentes que atuam no envio e recebimento de mensagens eletrônicas. Os protocolos, de um modo geral, são um conjunto de regras que define como os dados serão trafegados na rede, incluindo controle de erros e retransmissão. Os agentes de e-mail são responsáveis pelo envio e recebimento das

[1] *Request For Comments* - Todos os protocolos são padronizados e mantidos pela *Internet Engineering Task Force* (IETF), onde são publicados os RFCs.

mensagens eletrônicas, através da interação com o usuário. Os agentes trabalham em conjunto com os protocolos de envio e recebimento de mensagens eletrônicas.

Os agentes que compõem o e-mail são *Mail Transfer Agent* (MTA) e *Mail User Agent* (MUA), que correspondem, respectivamente, o servidor de e-mail e o agente de usuário. Há ainda dois agentes que agem entre o cliente e servidor: o *Mail Delivery Agent* (MDA), responsável pela entrega do e-mail e o *Mail Access Agent* (MAA), que permite o acesso do usuário às mensagens. A Figura 1 representa uma visão do funcionamento do sistema de e-mail.

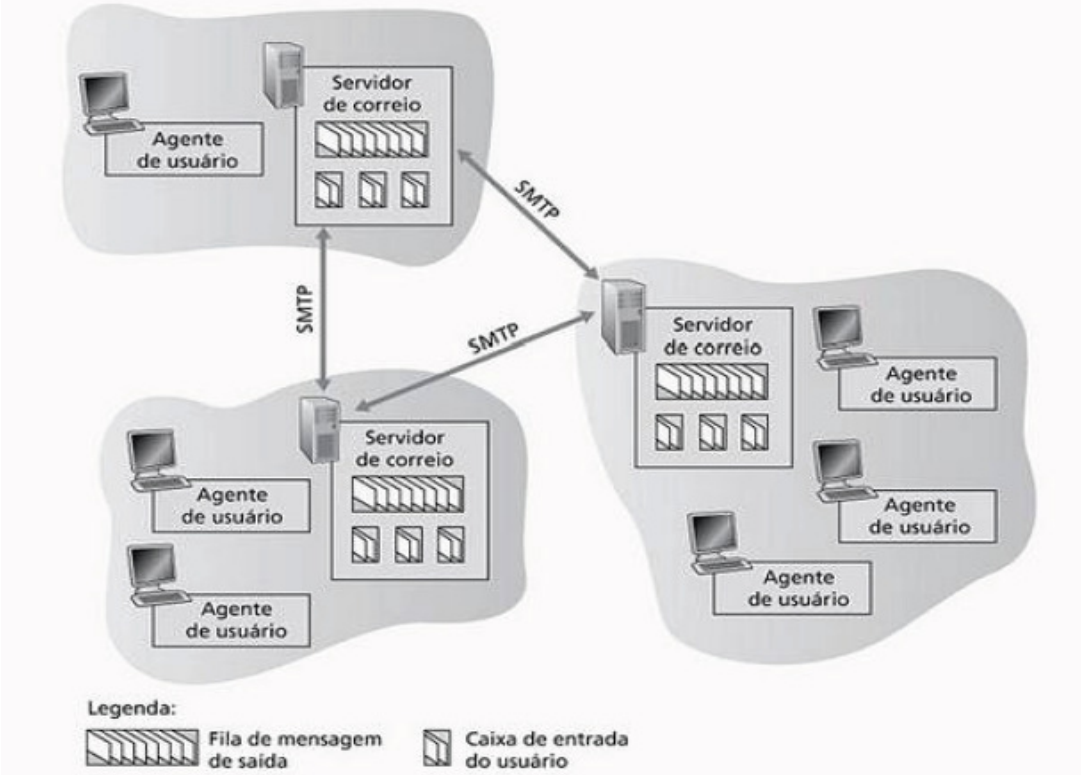

Figura 1 - Visão do sistema de e-mail (KUROSE e ROSS, 2010, p. 88)

A Figura 1 apresenta os três principais elementos para o envio e recebimento de mensagens eletrônicas, são eles: agente de usuário, agente responsável pelo servidor de e-mail e os protocolos. Tudo começa com o usuário ao enviar um e-mail, onde entra em funcionamento o agente de usuário que faz a ligação com o servidor de e-mail e este através do protocolo de envio (SMTP) envia para o servidor de e-mail de destino, ao chegar no servidor de e-mail destino, este faz o processo inverso do envio até chegar a caixa de entrada do usuário.

O e-mail possui uma estrutura simples, com campos bem definidos que se assemelham a uma correspondência manual. Os campos de e-mail são divididos em

cabeçalho, data, assunto e corpo do e-mail. Estes campos são apresentados por meio da Figura 2.

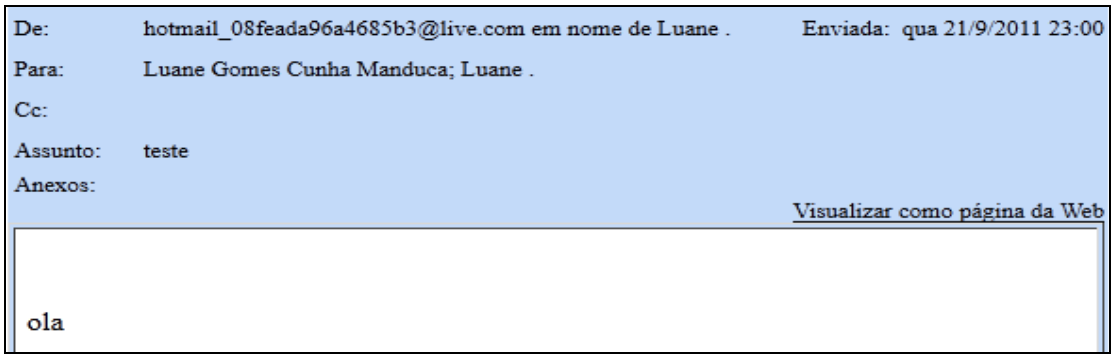

Figura 2 - Campos de um e-mail

Como visto na Figura 2, o cabeçalho é destinado à identificação do remetente (De) e destinatário (Para), o campo "Assunto" é reservado para o tema da mensagem e o "corpo" corresponde ao conteúdo do e-mail e por fim a data de envio do e-mail.

Para a comunicação entre servidores de e-mails foram estabelecidos protocolos para envio e recebimento de mensagens. Todos os protocolos são padronizados e mantidos pela *Internet Engineering Task Force* (IETF), onde são publicados os RFCs para cada protocolo. Os principais protocolos que operam sobre o e-mail são:

- **SMTP***: Simple Mail Transfer Protocol* é um protocolo da camada de aplicação que serve para o envio de mensagens via Internet baseado em texto simples e é padronizado pelo RFC 821 (1982). Porém, para Comer (1998, p. 490), o SMTP é bem mais complexo, devido a uma série de comandos dados pelo SMTP. Scrimger *et. all* (2002, p. 371), descrevem o protocolo em cinco passos, que vão desde ao estabelecimento da conexão ao recebimento do relatório sobre a mensagem enviada, assim os autores destacam o protocolo baseado em uma entrega ponta a ponta.
- **POP3**: *Post Office Protocol* versão 3, protocolo responsável por recuperar mensagens do servidor. Descrito na RFC 1939, que descreve o POP3 como um protocolo que permite que uma estação de trabalho recupere e-mails que está em um servidor. O protocolo passa por três estados: autorização, transação e atualização (KUROSE e ROSS, 2010, p. 94). Estas fases ocorrem de maneira sequencial ao acessar o servidor de e-mail, no qual há uma interação entre o agente de usuário e o servidor de e-mail; esta interação é interpretada através de comandos.
- **IMAP**: *Internet Message Access Protocol*, que possui as mesmas características do POP3, porém, com algumas melhorias como, por exemplo, "acessar as informações em

> computador remoto como se estivessem armazenadas no computador local" (SCRIMGER *et. all*, 2002, p. 372), além de possuir uma conexão sempre ativa com o servidor. Para Kurose e Ross (2010, p. 95), "Outra característica importante do IMAP é que ele tem comandos que permitem que um agente de usuário obtenha componentes de mensagens". Ou seja, o usuário pode fazer o *download* de apenas parte de uma mensagem que está no servidor, por exemplo, abrir somente o cabeçalho da mensagem. Tal característica é importante em ocasiões na qual o usuário esteja acessando com uma conexão de baixa velocidade.

Além dos protocolos descritos anteriormente, existe outro protocolo denominado MIME (*Multipurpose Internet Mail Extension*), que é uma extensão do SMTP com suporte para envio de mensagens multimídia. Recentemente, em março de 2011, foi lançado o RFC 6152, para o protocolo SMTP com suporte a envio de mensagens de 8 bits.

Os protocolos SMTP, POP3 e IMAP são utilizados por um cliente de e-mail, como, por exemplo, o *Microsoft Office Outlook.* Segundo Kurose e Ross (2010, p. 95), "hoje, um número cada vez maior de usuários está enviando e acessando e-mails por meio de seus browsers web". Neste caso, o protocolo utilizado é o HTTP (*HyperText Transfer Protocol*).

Devido à facilidade de utilização dos e-mails, ao baixo custo, entre outros benefícios advindos com o e-mail, um novo problema começou a surgir: usuários mal intencionados passaram a utilizar o e-mail como uma espécie de ferramenta para envio de propagandas, vírus, boatos, correntes etc.. A partir disto, um novo termo começou a surgir, *spam*, que será apresentado na próxima seção.

2.2. *Spam*

O termo *spam*, dentro do contexto da informática, é caracterizado como uma mensagem eletrônica não solicitada pelo destinatário, podendo ou não ser enviada para uma grande quantidade de pessoas (ANTISPAM.BR, 2005, online). O primeiro *spam* eletrônico é relatado na década de 70, quando um funcionário da empresa DEC (*Digital Equipment Corporation*) enviou uma mensagem aos usuários da Arpanet, convidando-os para lançamento de seus novos produtos (TAVEIRA *et. all*, 2006, p.05). Com a popularização do Internet e do e-mail, o número de envio de *spam* cresceu exponencialmente desde a década de 90 e atualmente passou a incorporar novos ambientes, como as redes sociais, SMS (*Short Message Service*) e o VoIP (*Voice over IP* – Voz sobre IP).

Por exemplo, dados da Symantec, em seu Relatório *Intelligence* de julho de 2011, relatam que apesar do nível global de envio de *spam* ter diminuído neste mês, a quantidade de *spam* teve um crescimento em relação ao mês de junho, representando um aumento de 4,9% em comparação com o mês anterior. No Brasil, o envio de *spams* representou no mês de julho de 2011 78,7% do tráfego de e-mails (Symantec Intelligence Report, 2011, p. 9). Segundo dados da *AVG Community Powered Threat* (2011, online), no terceiro trimestre deste ano o Brasil foi responsável por 5,41% dos e-mails enviados em todo o mundo, ficando em terceiro lugar entre os países que mais enviam *spams*, perdendo apenas para os EUA e a Índia.

As consequências do envio de *spam* vão desde os prejuízos econômicos a *stress* causado pela perda de tempo que se leva para selecionar e excluir as mensagens indesejadas. Os prejuízos econômicos são referentes a perca da largura de banda (o que consequentemente pode retardar o recebimento de e-mail legítimo), financeiros (e-mails que contenham algum tipo de golpe), além do investimento gasto para conter o número de *spams*. Os responsáveis pelo envio de tais mensagens são chamados *spammers*.

Existem *spams* com diversos intuitos e esses são catalogados segundo a sua finalidade. As principais categorias de *spams* são apresentadas na próxima seção.

2.2.1. Categorias de *Spams*

Os *spams* são catalogados de acordo com o seu propósito, sendo que os mais comuns são aqueles destinados a fazer propaganda de algum produto ou serviço. De acordo com M86 Security Labs (2011, online), entre essas propagandas existe uma que se destaca pela quantidade, são as de produtos farmacêuticos, que representam 43,65% de *spams* enviados.

Existem ainda diversos outros tipos de *Spams*, que vão desde a um simples boato aos que são destinados para roubar dados do usuário. Os principais tipos de *spams* existentes serão listados a seguir (M86 SECURITY LABS, 2011, online):

- **farmacêutica:** como já foi dito, este tipo de *spam* tem o intuito de divulgar produtos farmacêuticos, tais como: pílulas, fórmulas, manipulados, ervas, etc.. Esses tipos de *spams* divulgam também os possíveis benefícios com o uso da droga divulgada. Os mais comuns são relacionados à estética (perca peso, ganhe músculos, tratar calvície, entre outros);
- ***scams*:** *spams* com conteúdo extenso que tentam confundir ou/e enganar o usuário no intuito de "pescar" informações, ou seja, o e-mail serve como uma isca para coletar informações. Geralmente são e-mails com algum tipo de promoção ou suposto prêmio e

para participar da promoção ou receber o prêmio, o destinatário precisará acessar uma página e nesta será pedido algumas informações do destinatário.

- **adulto:** *spams* com conteúdo pornográfico ou associação a sites que contenham tal conteúdo. Podem-se incluir também neste tipo de *spam* os serviços referentes a namoro online e anúncios sobre este.
- **financeiro:** conteúdo relacionado a ofertas de créditos, seja para empréstimo ou financiamento.
- ***phishing*:** este é um dos mais perigosos tipos de *spam*, pois são *e-mail*s destinado a roubo de dados do usuário. O e-mail é enviado ao destinatário como se fosse um e-mail legítimo levando o usuário a clicar em um link, que supostamente seria, por exemplo, do banco em que o usuário possui conta, mas na verdade ele é direcionado a uma falsa página para que o usuário digite a dados referentes a sua conta e assim tenha suas informações coletadas.
- **educação:** oferece qualificação profissional de nível superior, técnico e até mesmo cursos rápidos, como, por exemplo, um curso de atendente de escritório.
- **propagandas:** e-mail contendo algum tipo de anúncio, divulgação ou promoção de um produto ou serviço.
- ***malware*:** *spams* com anexos ou com links direcionados a sites de códigos maliciosos. Geralmente são e-mails solicitando que o usuário faça algum tipo de atualização, por meio de algum arquivo em anexo para ser executado. Esse tipo de *spam* tem o objetivo de infectar a máquina do usuário a fim de danificar *softwares* ou arquivos e roubar dados através de programas denominados *keyloggers*.
- **Jogo:** oferece promoções ou bônus para jogos online. Os mais comuns são relacionados a cassinos online.

Além destes citados, há outro tipo de *spam* muito comum, são os e-mails conhecidos como correntes e boatos. Na verdade, esta forma serve como técnica que os *spammers* utilizam para alimentar seu banco de dados de e-mails. Por exemplo, o *spammer* envia uma corrente para algumas pessoas, e estas, ao receberem, encaminham para os seus contatos (incluindo o remetente -*spammer*), e estes contatos encaminham para os seus contatos e assim o envio vai crescendo rapidamente. Isso acontece, pois normalmente quando as pessoas vão encaminhar uma mensagem, não limpam os endereços remanescentes e também não utilizam o campo Cc, copia oculta, que serve para que os destinatários não visualizem os endereços de

e-mails de outras pessoas. Sem tomar estas devidas precauções, fica fácil de alguém mal intencionado usar os endereços de e-mails indevidamente.

Os *spammers* também utilizam diversas outras técnicas para obtenção de e-mails, umas delas é denominada *harvesting*, que segundo Antispam.br(2005, online) "é uma técnica, utilizada por *spammers*, que consiste em varrer páginas *Web*, arquivos de listas de discussão, entre outros, em busca de endereços de e-mail". Outra técnica utilizada pelos *spammers* também é a previsibilidade, no qual os *spammers* utilizam os nomes mais comuns para alimentar a sua lista de endereços de e-mails. Certamente já existem e-mails com os nomes mais comuns em servidores de e-mails gratuitos, por exemplo, fabiano@dominio.com, neste exemplo domínio.com deve ser substituído por qualquer servidor gratuito de e-mail.

Com a diversidade de tipos de *spams* existentes e os prejuízos causados pelo envio destes, empresas de seguranças desenvolvem e usam técnicas de detecção de *spams* na tentativa de conter o envio e/ou impedir que um *spam* chegue à caixa de entrada do usuário. A próxima seção trata as principais técnicas de detecção de *spams*.

2.2.2. Técnicas de detecção

Técnicas de detecção de *spams* se resumem a sistemas anti-*spam*, que segundo Taveira (2008, p.11), estas têm como objetivo "reduzir o número de *spams* recebidos por um usuário, classificando as mensagens para, então, filtrá-las". Ainda, segundo Szendrodi e Bandeira (2005, online) "técnicas ANTI-*SPAM* destinaram-se a preencher as lacunas que a simplificação do protocolo SMTP deixou evidente".

Existem duas principais técnicas de detecção de *spams*, as técnicas baseadas em filtros e as técnicas baseada em conteúdo. As técnicas baseadas em filtros são subdivididas entre filtros manuais e dinâmicos, sendo que estes ainda se subdividem em outras categorias. Os filtros definidos pelos usuários (manuais) são os mais simples de serem configurados, porém, mais trabalhosos, pois requerem uma intervenção humana, já que o usuário tem que selecionar a mensagem e reportar como *spam* para o servidor de e-mail. Tal filtro acaba se tornando um filtro personalizado, pois foi configurado de acordo com as definições do usuário. Esta é apenas uma das muitas opções existentes para bloquear *spams*, existe um conjunto de ferramentas e técnicas que se combinam na tentativa de bloquear e-mails indesejados, como por exemplo, os filtros dinâmicos.

Os filtros automatizados possuem duas principais categorias, que são os filtros baseados em listas de bloqueio e filtros baseado em aprendizagem de maquina. As listas de bloqueio são chamadas de lista negra e a lista de liberação, lista branca. Os primeiros sistemas

anti-*spam* eram baseados em listas, tais listas são chamadas de lista branca e lista negra. A lista branca contém os e-mails que sempre serão liberados, pode conter também uma lista de domínios permitidos, que são tidos como confiáveis. Já as listas negras, *blacklists,* são domínios bloqueados para recebimento de e-mails, se um servidor cair em uma lista dessa, todos os usuários serão bloqueados. Essas listas também são chamadas de DNSBL (*Domain Name System Blacklist*) fazem o bloqueio de e-mails através de consultas DNS. Segundo Taveira (2008, p. 13) "verificação é realizada através do envio de um pedido DNS para um servidor DNS que fará a consulta à lista DNSBL". A inclusão de endereços é automática através da verificação de servidores que enviam e-mails indiscriminadamente. Porém, se algum servidor tiver o seu domínio incluído erroneamente, como, por exemplo, em decorrência de algum vírus, a exclusão deste servidor da lista de bloqueio será feita de forma manual, que pode demorar, causando o bloqueio de e-mails válidos.

Além dos filtros baseados em listas, existem outras categoria de filtros baseados em técnicas de aprendizado de máquina. Nesta categoria estão incluídas as redes neurais, redes bayesianas e os Sistemas Imunológicos Artificiais. Estas técnicas passam por uma fase de aprendizado, no qual irão "aprender" a diferenciar um e-mail de um *spam*. Estas técnicas se baseiam no conteúdo do e-mail, por isso são chamadas de filtros de conteúdo. Tais filtros se baseiam na probabilidade do conteúdo de um determinado e-mail ser considerado um *spam*, com base no aprendizado que é realizado. Algumas técnicas de filtros baseados em conteúdo são:

- filtro bayesiano: o filtro é baseado em uma rede bayesiana que Segundo Russell e Norvig (2004, p. 480) "é um grafo orientado em que cada nó é identificado com informações de probabilidade quantitativa". O grafo que forma a rede é direcionado e acíclico, na qual cada nó representa uma variável e as arestas representam a ligação entre as variáveis. Tal ligação é de causa e efeito, ou seja, X interfere em Y e Y está condicionado a X, como pode ser observado na Figura 3.

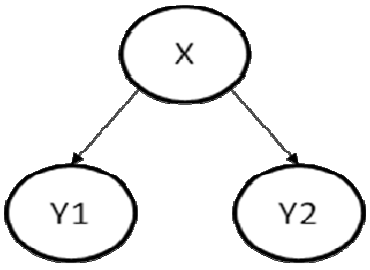

Figura 3 - Representação simples de uma Rede Bayesiana

- Sistema Imunológico Artificial: filtros baseados nesta técnica são construídos a partir de uma experiência anterior, na qual já se tenha alimentado a base de conhecimento que servirá de base para as próximas verificações. Como dito, esta técnica possui características dinâmicas, de forma que a escolha do algoritmo utilizado baseado na técnica para a utilização do bloqueio de *spams* deve ser de acordo com as características de cada algoritmo juntamente com o resultado desejado (a seção 2.4 lista os principais algoritmos desta técnica). Basicamente, o bloqueio de *spam* será feito através de aprendizagem e memória, ou seja, o algoritmo aprende e é capaz de "lembrar" de um determinado fato e assim poder classificar. Carvalho (2009, p. 63) faz uma analogia do *spam* com o modelo imunológico biológico, no qual o filtro é representado em duas fases: a primeira corresponde ao modelo inato, que possui um conjunto fixo de elementos, realizando a primeira barreira; e a segunda ao adaptativo, que conta com o reconhecimento de padrões das células T e B, que realizam a seleção negativa, reconhecendo o próprio, representados por e-mails legítimos, e o não-próprio, e-mails considerados *spams*.

Ferramentas como SpamAssassin e o Bogofilter utilizam a técnica baseada filtro bayesiano. O Bogofilter separa o e-mail recebido em *tokens*, ou seja, padrões de palavras que serão comparados com uma base de dados, que contém a quantidade de vezes que tal padrão apareceu em e-mails. O resultado dessa comparação irá avaliar a probabilidade do e-mail ser um *spam* ou não, a partir de probabilidades apoiadas na teoria bayesiana (ZUCCO, 2005, p. 21).

O SpamAssassin, que pode ser aplicado tanto no servidor quanto na máquina cliente, adota vários mecanismos para identificar *spams*, entres eles o filtro bayesiano. O SpamAssassin trabalha com um conjunto de regras que são usadas para determinar se um e-mail é um *spam*. Além dessas ferramentas, existem outras que também usam a teoria bayesiana com parte do filtro, são elas: DSPAM e CRM114.

Existe outra técnica de detecção de *spam* que é baseada em reputação. A reputação de um e-mail pode ser medida através de redes sociais, no qual cada usuário é representado como um nó na rede e o envio de mensagem representa uma aresta; autenticação de remetentes ou reputação dos servidores, ou seja, o quanto um determinado endereço de e-mail é confiável ou não, classificando-o como um *spam* de acordo com a sua reputação.

Taveira (2008, p.38) propôs um mecanismo de defesa de spams baseado em autenticação e reputação, no qual cada servidor avalia a reputação dos outros servidores, a reputação de um servidor é determinada pelo quantitativo de mensagens spams que foram

enviadas a partir deste servidor, nesta classificação é utilizado um anti-*spam* convencional, no qual tem um peso maior para medir a reputação as ultimas mensagens ao invés do total de mensagens enviadas, pois um servidor pode ser confiável em um momento e em outro não.

O principal problema da detecção de *spams* são os falsos positivos, ou seja, uma mensagem válida filtrada como *spam*, o que acarreta o não recebimento de uma mensagem legítima pelo usuário. Há também o falso negativo, que é um *spam* legítimo, mas o usuário recebeu como um e-mail válido, ou seja, na sua caixa de entrada, neste caso, o dano não é tão grave, já que o usuário pode detectar que é um *spam* e então descartar a mensagem.

Das técnicas de detecção de *Spams* apresentadas neste trabalho, a mais utilizada atualmente são os filtros bayesianos, que com a junção com outras técnicas e o treinamento da base de dados proporcionam filtros cada vez melhores. Porém, assim como os métodos anti-*spams* evoluem, os *spammers* também se atualizam, o que cria a necessidade um método adaptativo que acompanhe a mutação dos *spams*. Tal solução pode ser dada através de métodos adaptativos como o SIA, que deve proporcionar detecção baseada em memória e experiência, sendo capaz de detectar um e-mail como *spam* mesmo sem nunca ter recebido tal e-mail até o momento.

Dentro das possíveis áreas de aplicação do Sistema Imunológico Artificial, a mais empregada é a referente à segurança[2], pois os princípios da resposta imunológica adaptativa caracterizam-se por ser uma resposta mais eficiente, especialmente, após a maturação das células e estas apresentam características como memória, tolerância a falhas, reconhecimento do próprio e não-próprio. Coppin (2010, p. 330) destaca quatro áreas de aplicação do SIA, são elas: segurança de computadores, busca combinatória, aprendizado de máquina e detecção de falhas, ressalta ainda que o estudo sobre SIA é um campo relativamente novo. Diversos autores já comprovaram a eficácia da técnica, destacando-a pela a sua capacidade de reconhecimento e adaptação.

Este trabalho irá focar na detecção de *spams* que se encontra dentro da área de segurança computacional, como ressalta Guzella *et. All* (2006, p.02) "muitos daqueles que enviam tais mensagens chegam a até mesmo invadir servidores de e-mail vulneráveis, usando-os para enviá-las, o que dificulta o seu rastreamento e efetivamente caracteriza uma invasão". Os *spams* também representam uma fonte de disseminação de vírus e demais "pragas virtuais" que podem danificar *softwares* ou arquivos e ainda roubar dados de usuários. Nas seções

[2] Para Kurose e Ross (2010, p. 493), a segurança em rede possui as seguintes propriedades: confidencialidade; autenticidade; integridade e segurança operacional.

seguintes serão abordados conceitos sobre o Sistema Imunológico Natural e suas características.

2.3. Sistema Imunológico Natural

O Sistema Imunológico Natural é o sistema de defesa do corpo, formado por células, órgãos e moléculas que trabalham de modo harmônico para que o corpo seja defendido contra ameaças de bactérias, vírus ou fungos. Além de possuir interação entre si, o sistema imunológico trabalha em conjunto com os demais sistemas do corpo humano, sendo que seu principal aliado é o sistema linfático, que é o responsável pelo transporte de fluidos linfáticos através dos vasos linfáticos, contribuindo também para a circulação e renovação do sangue (FERRON e RANCANO, 2007, p. 57).

As principais características do sistema imunológico são (ALMEIDA *et all*, 2007, p. 139):

- unicidade: cada sistema é único;
- reconhecimento de padrões internos e externos ao sistema: permite que células invasoras sejam reconhecidas e eliminadas;
- detecção de anomalia: referente à capacidade de reconhecer agentes causadores de doenças, mesmo nunca tendo sido exposta a tal agente;
- detecção imperfeita: não há necessidade de um reconhecimento perfeito do agente causador de doenças para que o Sistema Imunológico entre em ação;
- diversidade: sendo que uma quantidade limitada de células é utilizada para o reconhecimento de um número infinito de elementos, inclusive os de laboratório;
- aprendizagem por reforço: sendo que a cada contato com o patógeno, o sistema imunológico aperfeiçoa a capacidade de resposta;
- memória: capacidade que o sistema possui de armazenar o reconhecimento de ataques, para uma resposta mais rápida e eficaz.

O Sistema Imunológico se divide em dois tipos: o inato e o adaptativo. O inato é aquele em que o indivíduo já nasce com ele, sem a necessidade de indução de nenhum tipo de agente. Este tipo de Sistema Imunológico não apresenta a característica de reação de acordo com o agente infeccioso, ou seja, a resposta imune é a mesma a qualquer tipo de invasão e a resposta a um determinado agente é imediata. Já o sistema imune adaptativo ou adquirido, possui características mais dinâmicas, tais como: memória, seleção e evolução. Suas

principais células são os linfócitos que realizam a seleção clonal, apresentada por Frank Macfarlane Burnet em 1959, como um processo em que somente as células capazes de reconhecer antígenos[3] irão se reproduzir. Na imunidade adquirida o indivíduo passa por um processo de evolução celular em que as células vão adquirindo resistência através da exposição ao antígeno que se dá de maneira natural ou induzido através da vacinação. Segundo Parham (2001, p. 27) "A vacinação é um modo de estimular a imunidade protetora administrando os antígenos de um patógeno em uma forma que não provoque a doença".

O Sistema Imunológico é estruturado em camadas e está presente desde a superfície da pele, através da própria pele, até chegar à resposta imune adaptativa. Esta estrutura não é executada sequencialmente, podendo ser gerada somente uma resposta imune inata, através do processo de fagocitose, por exemplo. . Fagocitose é processo em que células fagocitárias englobam agentes causadores de doenças, eliminando-os através de um processo de digestão celular ou através da autólise. A Figura 4, a seguir, apresenta a estrutura do sistema imunológico em multicamadas.

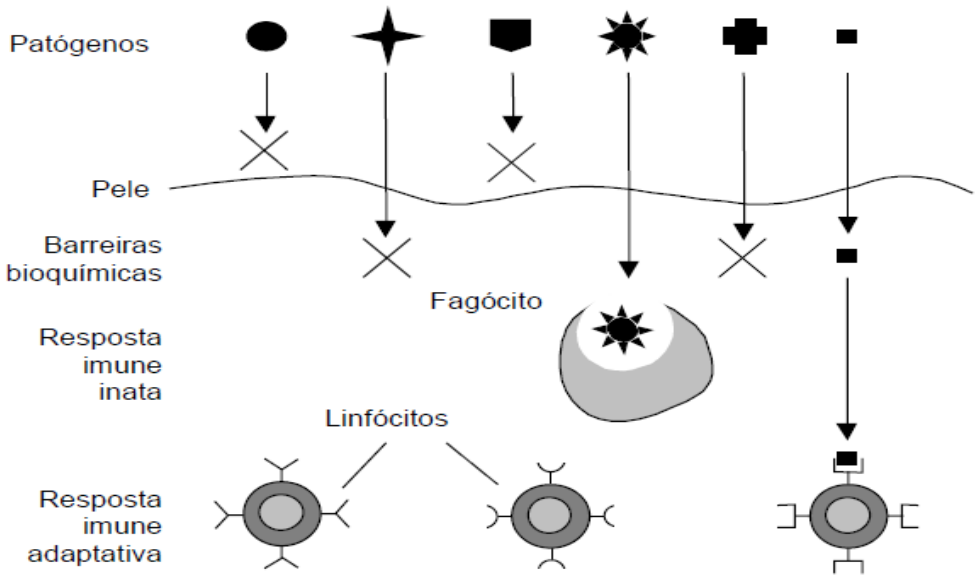

Figura 4 - Estrutura multicamadas do Sistema Imunológico (CASTRO, 2001, p. 16)

Na estrutura multicamadas do Sistema Imunológico, podem-se observar os patógenos, causadores de doenças, ultrapassando a primeira barreira, a pele, passando pela barreira bioquímica (fluídos, suor, lágrimas) até chegar à resposta imune inata, onde ocorre a fagocitosePor fim a ultima barreira do Sistema Imune que é resposta imune adaptativa.

[3] Substância que, introduzida no organismo, provoca a formação de anticorpos específicos (AULETE, 2011, Online). Provoca uma resposta imunológica.

O Sistema Imunológico Adaptativo possui características que são desejáveis na área da computação, como por exemplo: adaptação e memória. Por isto, a próxima seção dará ênfase a este tipo de sistema, bem como apresentar estas características.

2.3.1. Sistema Imunológico Adaptativo

O Sistema Imunológico Adaptativo é formado, principalmente, por um conjunto de células, que possuem a característica de se adequarem a determinada situação, o que forma a resposta imunológica adaptativa. As principais células do Sistema Imunológico são os linfócitos, que se dividem em duas categorias, chamadas de células B e células T (CASTRO, 2001, p. 17).

As células B ou linfócitos B ainda são subdivididos em dois tipos: os plasmócitos e as células de memória. Porém, antes dessa divisão, acontece a seleção clonal que será apresentada na seção 2.4.1. Estas células são responsáveis pela imunidade humoral, ou seja, sua principal função é a produção de anticorpos que se dá através do reconhecimento do antígeno. Após o reconhecimento do antígeno, a célula B pode se transformar em plasmócitos, que são células plasmáticas capazes de combater infecções. As células B, que não se transformam em plasmócitos, viram células de memória, que circulam pelos vasos linfáticos e são capazes de guardar informações. Tais informações servem para uma resposta mais efetiva a um ataque posterior. Um exemplo disso acontece com doenças como caxumba, catapora etc., em que o indivíduo só é contaminado somente uma vez durante seu ciclo de vida.

Já as células T, ou linfócitos T, também se subdividem em vários tipos de células; as principais são as células auxiliares, conhecidas como Th (*helper*), e as células reguladoras, que realizam a imunidade celular. As células Th auxiliam outras células, interagindo com elas, como, por exemplo, a interação com a célula B, onde o Th estimula o processo de produção de anticorpos, atuando como um alerta, “avisando” o sistema imunológico quando há algum ataque. Assim, quando esta célula é atacada como, por exemplo, pelo vírus HIV, que age diretamente nessa célula, o sistema imunológico fica sem alerta de defesa, deixando todo o corpo mais vulnerável a ações de patogênicos.

Já as células regulatórias são responsáveis pela identificação do próprio e não-próprio, ou seja, são capazes de diferenciar o que é próprio do corpo do que não é. Porém, antes que essas células sejam capazes de fazer tal reconhecimento, elas passam por uma seleção, chamada de seleção positiva e seleção negativa. A seleção positiva seleciona as células capazes de operar em uma resposta imunológica; estas são chamadas de células

imunocompetentes (CASTRO, 2001, p. 32). Já a seleção negativa seleciona e elimina as células que não são capazes de reconhecer as células próprias do corpo. Tal seleção será explicada na seção 2.4.3, por ser importante para o reconhecimento do próprio e não-próprio, ou seja, o reconhecimento de padrões.

O reconhecimento próprio é a capacidade que as células possuem de identificar os organismos pertencentes ao corpo, e a identificação do não-próprio permite o reconhecimento de agentes infecciosos. Tal reconhecimento é importante para as células de defesa não combaterem células do corpo humano. Um exemplo desse reconhecimento acontece após o transplante, quando o sistema imunológico deve possuir a capacidade de reconhecimento de um novo órgão presente no corpo humano e, quando esse reconhecimento não acontece, ocorre o que é conhecido como rejeição.

Tanto as células B quanto as células T possuem uma característica importante, que é o reconhecimento de padrões, ambas são capazes de reconhecer antígenos. Só que há uma importante diferença no reconhecimento: enquanto as células B fazem o reconhecimento dos antígenos puros, livres de qualquer solução, as células T fazem o reconhecimento dos antígenos peptídicos (processados) ligados ao complexo principal de histocompatibilidade - MHC (*Major Histocompatibility Complex*). O reconhecimento realizado pela célula B é feito através de um receptor chamado de BCR (B *cell receptor*), capaz de reconhecer uma infinidade de antígenos, já o receptor da célula T é chamado de TCR (T *cell receptor*) (CASTRO, 2001, p. 18).

Baseada nas informações contidas até aqui, a próxima seção aborda o Sistema Imunológico Artificial, tendo como cerne o Sistema Imunológico Adaptativo, pois como já foi dito, tal sistema possui características desejáveis em sistemas computacionais. Serão apresentadas também, nas subseções, teorias do Sistema Imunológico Natural que validam a implementação de um Sistema Imunológico Artificial. Essas teorias são: Seleção Clonal, Teoria da Rede Imune e Seleção Negativa.

2.4. Sistema Imunológico Artificial

O estudo e a implementação de sistemas artificiais inspirados na natureza tem se intensificado ultimamente, graças às tecnologias que possibilitam tal desenvolvimento, juntamente com a necessidade de se aplicar esses conceitos a diversas áreas. Em geral, de acordo com Von Zuben (2011, p. 04), o termo artificial: “é tudo aquilo que é feito pelo ser humano, ou seja, um artefato”. Partindo para um conceito mais específico, ainda segundo Zubem (2011, p. 5), artificial é algo produzido pelo homem com algum propósito, que imita ou não aparência do

natural. Esse material artificial produzido pode ser utilizado como objeto de estudos como, por exemplo, um coração artificial para ser estudado por alunos de medicina, ou, ainda, seus conceitos podem ser aplicados em sistemas computacionais.

Os Sistemas Imunológicos Artificiais, segundo Dasgupta (1998 *apud* ALMEIDA *et all*, 2007, p. 139), "são mecanismos computacionais compostos por metodologias inteligentes, inspiradas no sistema imunológico biológico, para a solução de problemas do mundo real". Tais problemas poderão ser solucionados, ou ainda otimizados, com a utilização do Sistema Imunológico Artificial, pois já se tem conhecimento de diversas áreas de aplicações baseadas em tal sistema, como segurança, robótica, reconhecimento de padrões, entre outras.

Para Amaral (2006, p. 14), os Sistemas Imunológicos Artificiais podem ser classificados em três categorias principais:

1. inspirados na teoria de Redes Imunológicas;
2. que são baseados no princípio da Seleção Clonal;
3. sistemas que utilizam técnicas inspiradas no mecanismo de reconhecimento próprio/não-próprio – Seleção Negativa;

A partir dessas categorias foram criadas metáforas do sistema imune, que são: Rede Imunológica, Seleção Clonal e Seleção Negativa (AMARAL, 2001, p. 47), já outros autores abordam como modelo. As categorias citadas do Sistema Imunológico Artificial são as mais utilizadas; existem outras teorias para o desenvolvimento de um Sistema Imunológico Artificial, mas estas são consideradas as principais e serão mais detalhadas nas seções posteriores.

Segundo Berbert (2008, p. 38), a "área de pesquisa dos Sistemas Imunológicos Artificiais é extensa. A escolha dos melhores modelos (Seleção Negativa, Teoria de Rede Imunológica, Seleção Clonal) depende do objetivo e das características do problema a ser estudado". Ou seja, dentro do Sistema Imunológico Artificial existem várias possibilidades a serem exploradas em diferentes problemas do mundo real. Os modelos citados acima, rede imunológica, seleção clonal e seleção negativa, são áreas de estudo dentro do Sistema Imunológico Artificial. Para Silva (2009, p.11), "o estudo das teorias dos sistemas imunes gera muitas possibilidades de abordagens de problemas em sistemas com características adaptativas ou inteligentes". A tabela 1 apresenta os modelos de Sistemas Imunológicos mais estudados, o problema e a área da aplicação.

Tabela 1- Resumo dos modelos de Sistemas Imunológicos Artificiais mais estudados (DASGUPTA, 2006, apud BERBERT 2008, p. 40).

Característica do Sistema Imunológico Biológico	**Problema Computacional**	**Aplicações Típicas**
Reconhecimento de próprio e não-próprio Detecção de mudanças	Detecção de mudanças ou anomalias	– Segurança de computador – Detecção de Falta
Teoria de Rede Imunológica e Memória Imunológica	Aprendizagem (supervisionada ou não-supervisionada)	– Classificação – Clusterização – Análise de dados – Mineração de dados
Seleção Clonal	Busca, Otimização	– Otimização de funções
Mobilidade e Distribuição	Processamento Distribuído	– Arquiteturas de agente – Controle Robótico – Descentralizado
Imunidade Inata	Teoria do Perigo	– Segurança de redes

Os modelos citados na Tabela 1, entre eles Teoria da Rede Imunológica, Seleção clonal e Seleção Negativa, são os mais utilizados para o desenvolvimento de um Sistema Artificial. Devido à complexidade de Sistema Imunológico Humano, o desenvolvimento de um SIA é composto, na maioria das vezes, por apenas um modelo, podendo também ser desenvolvido com a junção de outro modelo, assim, cada um desses modelos já foram estudados de forma individual por vários autores e em diferentes aplicações, considerando apenas a parte estudada e não o Sistema Imune como um todo. As seções a seguir apresentarão os conceitos dos principais modelos, bem como as características dos algoritmos baseados em cada modelo.

2.4.1. Teoria da Seleção Clonal

Como dito na seção 2.3, o processo de Seleção Clonal, apresentado por Burnet em 1959, é caracterizado pela capacidade que as células que produzem anticorpos têm de se reproduzirem. A Figura 5 apresenta a Teoria da Seleção Clonal, na qual há a seleção das células que foram capazes de reconhecer o antígeno. Logo após a seleção, essas células são clonadas e em seguida passam por um processo de diferenciação, transformando-se em células de memórias ou plasmócitos.

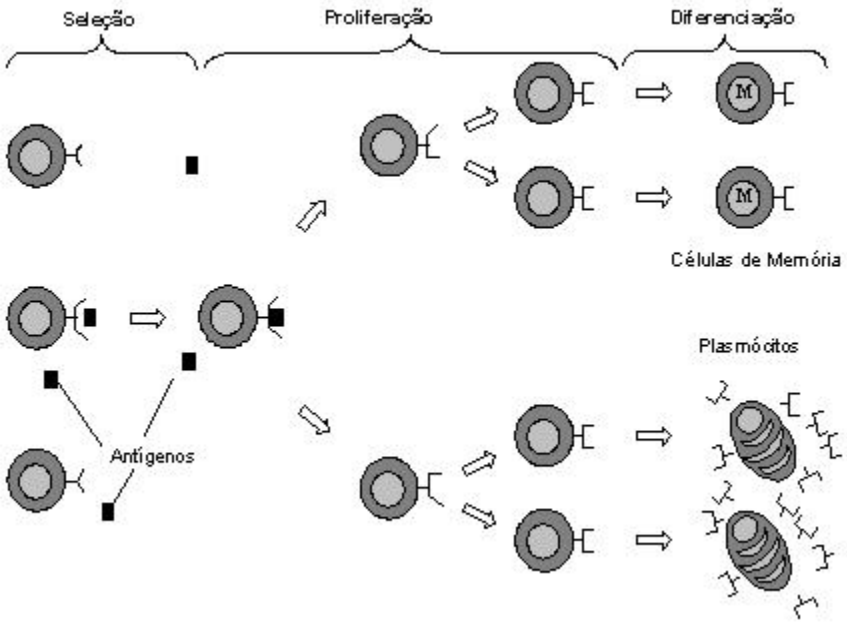

Figura 5 - Teoria da Seleção Clonal

Baseados na resposta adaptativa dada pela Teoria da Seleção Clonal, Castro e Von Zuben apresentaram em 2000 o Algoritmo de Seleção Clonal (*clonal selection algorithm*-CSA) com o propósito de realizar análise combinatória e otimização, como exemplo, o problema do caixeiro viajante, em que o viajante deve visitar cada cidade de um determinado território e depois voltar a cidade de origem com o menor custo possível. No ano seguinte, Castro e Von Zuben (2001, *online*) desenvolveram um novo algoritmo denominado CLONALG (*clonal selection algorithm*), no qual foi proposto, inicialmente, aprendizagem de máquina e reconhecimento de padrões e depois adaptado para solucionar problemas relacionados à otimização. O funcionamento do algoritmo será descrita a seguir, de forma genérica (CASTRO, 2001, p. 67):

1. gere um conjunto (P) de candidatos a solução, composto pelo subconjunto de células de memória (M) mais o restante (P{r}) da população ($P = P\{r\} \cup M$);
2. determine (processo de seleção) os n melhores indivíduos (P{n}) da população (P), baseado em uma medida de afinidade;
3. reproduza (processo de clonagem) estes n melhores indivíduos, gerando uma população temporária de clones (C). A quantidade de filhos de cada indivíduo é diretamente proporcional à sua afinidade;
4. submeta a população de clones a um esquema de hipermutação, em que a taxa de mutação é proporcional à afinidade do anticorpo. Uma população de anticorpos maduros é gerada (C*);

5. re-selecione os melhores indivíduos de C* para compor o conjunto de memória M;
6. substitua *d* anticorpos por novos indivíduos (diversidade). Os anticorpos com menores afinidades possuem maiores probabilidades de serem substituídos.

Todo esse processo entra em um laço de repetição infinito, dado que ao final do passo 4 dá-se início novamente à seleção das células maduras que serão re-selecionadas, formando a célula de memória, onde aquelas que tiverem menos afinidade serão substituídas. Tal processo caracteriza outro fator importante do sistema imunológico: a maturação e afinidade, que é caracterizado pela interação da célula com o antígeno e, a cada repetição dessa interação, a célula ganha maturidade e aumenta o nível de afinidade com o antígeno.

A seção seguinte apresenta a Teoria da Rede Imunológica.

2.4.2. Teoria da Rede Imunológica

Proposta originalmente por Niels Kaj Jerne, em 1974, a Teoria da Rede Imunológica, ou Rede Idiotípica, consiste em dizer que as células são capazes de reconhecer uma a outra, ou seja, uma célula é capaz de reconhecer outra e ser reconhecida, construindo assim uma rede de células que se auto-regulam, permanecendo em homeostase até que uma perturbação ocorra. Se alguma perturbação ocorrer, a rede se reorganiza para encontrar o equilíbrio novamente.

Castro & Von Zuben (2000, *online*) destacam duas características centrais da Rede Imunológica, a primeira é a capacidade de identificar indivíduos na rede e a segunda é a capacidade de aprendizagem de acordo com o ambiente em que está inserido.

A seguir, por meio da Figura 6, é demonstrada tal teoria, onde uma célula possui em suas extremidades o Paratopo e o Idiotopo. O Paratopo faz o reconhecimento do antígeno, já o Idiotopo é o responsável pelo reconhecimento de outra célula, formando assim a rede Idiotípica.

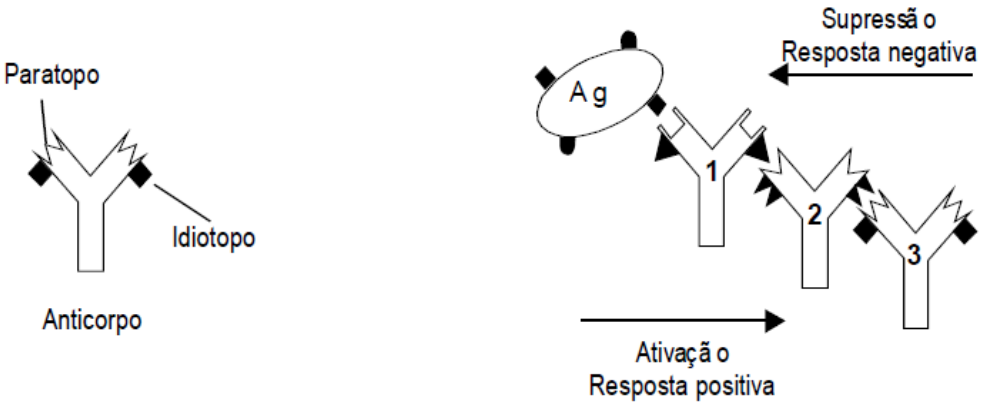

Figura 6 - Teoria da Rede Imunológica (CASTRO, 2001, p. 36)

Varela & Coutinho (1991, *apud* CASTRO & VON ZUBEN, 2000, *online*) destacam três características das redes imunológicas:

1. estrutura: descrição das interconexões levando em consideração elementos do sistema e não suas interações;
2. dinâmica: trata das interações entre os elementos do sistema;
3. metadinâmica: reestruturação da rede através da produção de novos anticorpos, mesmo que de forma sintetizada e a eliminação de células não estimuladas, toda essa reestruturação garante a capacidade do sistema imunológico continuar combatendo agentes nocivos ao corpo.

O algoritmo baseado nesta teoria aplica-se ao processo de análise de dados, como clusterização e classificação. A ferramenta aiNet (*Artificial Immune Network*), proposta por Von Zuben e Castro, e desenvolvida por Castro, é baseada nessa teoria. Utiliza ainda o algoritmo CLONALG para o processo de seleção e maturação. O aiNet é definido por Castro (2001, p.161) como:

> Um grafo com conexões ponderadas, não necessariamente totalmente interconectado, composto por um conjunto de nós, denominados anticorpos, e conjuntos de pares de nós chamados conexões, com um valor característico associado, chamado de peso da conexão ou simplesmente peso.

O algoritmo aiNet opera primeiramente com um processo de aprendizagem do antígeno em que o algoritmo realiza a interação de anticorpos pertencentes a rede a um antígeno e, posteriormente, é feita a qualificação da interação dos anticorpos da rede.

O algoritmo de treinamento realizado pelo aiNet é descrito da seguinte forma (CASTRO & VON ZUBEN, 2000, *online*):

1. A cada passo de iteração, faça:
   1.1. Para cada antígeno, faça:
      1.1.1. Determine a sua afinidade com as células de acordo com a métrica de distância adotada;
      1.1.2. Selecione as células com maior afinidade;
      1.1.3. Reproduzir (clone) as células selecionadas. Quanto maior a afinidade da célula, maior o número de clones gerados por cada célula estimulada;
      1.1.4. Aplicar a equação para os clones gerados;
      1.1.5. Determine dissimilaridade células;

1.1.6. Pegue as células selecionadas de maior afinidade e cria uma matriz (rede);

1.1.7. Eliminar as células cuja afinidade é inferior ao limite determinado, o que permite uma redução no tamanho da matriz;

1.1.8. Calcular a afinidade dos elementos da matriz;

1.1.9. Supressão clonal – eliminar as células que possuem menos afinidade entre si;

1.1.10. Concatenar a matriz contendo todas as células à matriz com as selecionadas;

1.2. Supressão da rede – substituição das células com menos interação por outras;

2. Teste o critério de parada.

Os passos 1.1 a 1.2 são repetidos até que o critério de iterações seja atingido. A figura a seguir, Figura 4, apresenta uma ilustração do algoritmo baseado na Teoria da Rede Imunológica.

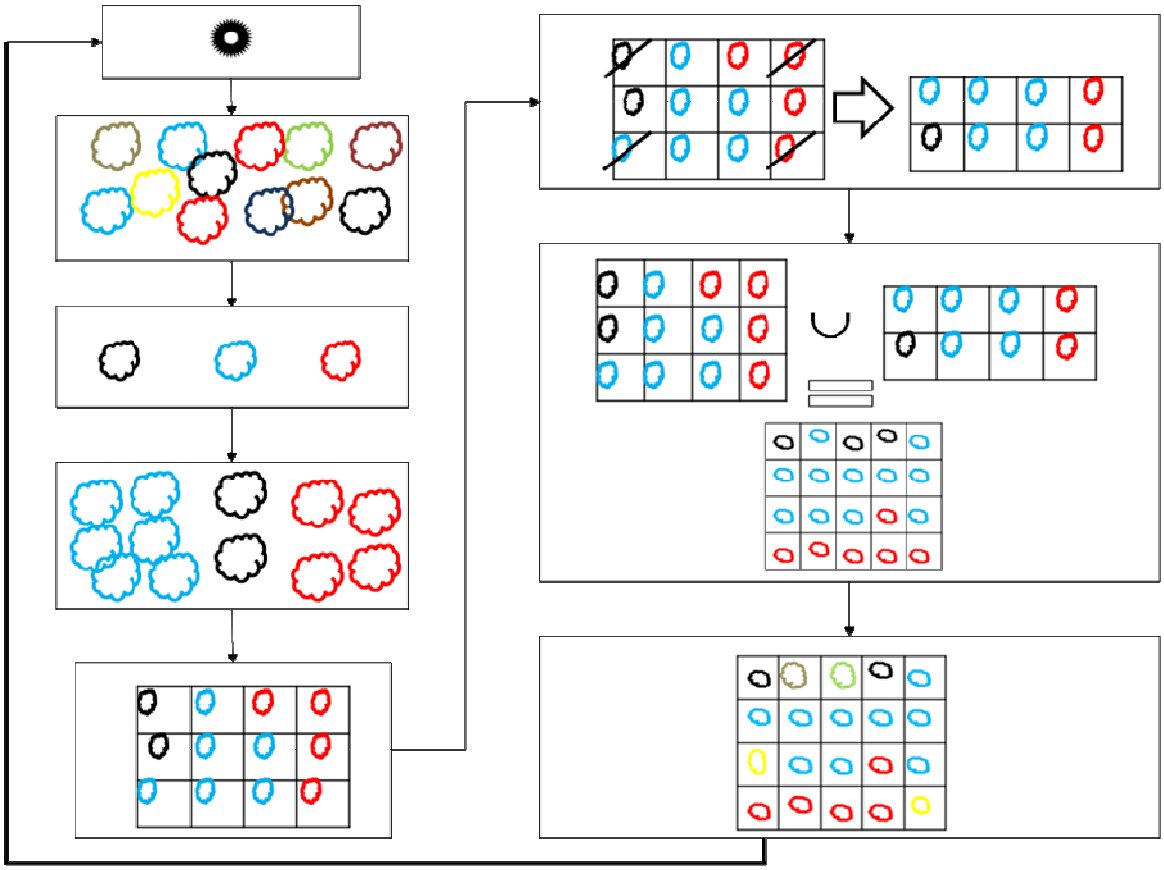

Figura 7 - Ilustração do algoritmo baseado na Rede Imunológica

A ilustração, apresentada na Figura 7, representa algumas etapas do algoritmo, no qual foi dividido em oito, sendo assim, a ilustração não contém todos os passos apresentados pelo algoritmo, contendo apenas os passos representativos. Tais passos serão descritos abaixo:

- o quadro 1 representa o passo 1.1 do algoritmo, no qual é apresentado o antígeno e iniciado um laço de repetição;
- no quadro 2 são ilustradas as células que irão interagir com o antígeno; tais células passarão por um cálculo que irá determinar as que serão clonadas;
- no quadro 3 são apresentadas as células selecionadas, que são as que tiveram maior afinidade com o antígeno e irão se proliferar;
- o quadro 4 apresenta os clones das células selecionadas, sendo que as que obtiveram maior afinidade foram capazes de gerar um maior número de clones;
- o quadro 5 exemplifica a formação da matriz (rede), onde uma célula liga-se a célula vizinha. Logo em seguida, o quadro 6 demonstra a eliminação das células por supressão da clonal, ou seja, eliminação das células com menos afinidade entre si, antes dessa supressão houve um cálculo de similaridade e ainda a eliminação das células com menor afinidade com o antígeno; estes dois últimos passos não estão demonstrados na figura, o resultado da eliminação das células forma uma nova matriz (rede);
- no quadro 7 é feita a concatenação da matriz que contém todas as células com a matriz que possui as células que passaram por todas as seleções;
- e, por fim, o quadro 8 apresenta a supressão da rede, ou seja, a eliminação de células com menos afinidade entre si, onde estas células serão substituídas por outras células aleatórias.

A próxima seção apresenta a Teoria de Seleção Negativa, conceito, características, fases do desenvolvimento.

2.4.3. Seleção Negativa

Segundo Amaral (2006, p. 47) “o objetivo do mecanismo de seleção negativa é fornecer tolerância às células próprias”. Esta seleção tem o objetivo selecionar as células que não são capazes de reconhecer as células próprias do corpo humano, eliminando as células que fazem esse reconhecimento, muito importante para que as células do corpo humano não sejam atacadas pelo sistema imunológico. Esta seleção acontece no Timo, que é um órgão do

sistema linfático responsável pela a maturação das células T. Após ocorrer a Seleção Negativa, as células T saem do Timo para realizar a defesa do corpo humano.

O mecanismo de Seleção Negativa se baseia na diferenciação das células próprias e não-próprias, *self* e *nonself*. Para Silva (2009, p. 15) *self/nonself* - é parte fundamental da teoria da Seleção Negativa. É uma teoria fundamentada no princípio de que o sistema imune funciona através da distinção de padrões conhecidos pelo organismo e, ao detectar um padrão desconhecido, a resposta imune é ativada.

Inspirado nessa seleção, o Algoritmo de Seleção Negativa (ASN), proposto em 1994 por Forrest, tem como base a geração de detectores, os quais detectam candidatos de modo aleatório, descartando aqueles que reconhecem dados relacionados, ou seja, os próprios, e tais detectores podem ser usados, mais tarde, para a detecção de anomalias (ALMEIDA 2006, p. 49). Por isto, o NSA é comumente usado na detecção de falhas ou anomalias, a proposta original do algoritmo é executada em duas fases, são elas (FORREST *et al.*, 1994):

1. Censoriamento: nesta fase são definidas e geradas as cadeias próprias;
2. Monitoração: avaliação da afinidade das cadeias com o conjunto de detectores.

A figura a seguir, Figura 8, ilustra a fase de Censoriamento, onde caracteriza a geração do conjunto de detectores e avaliação dos mesmos.

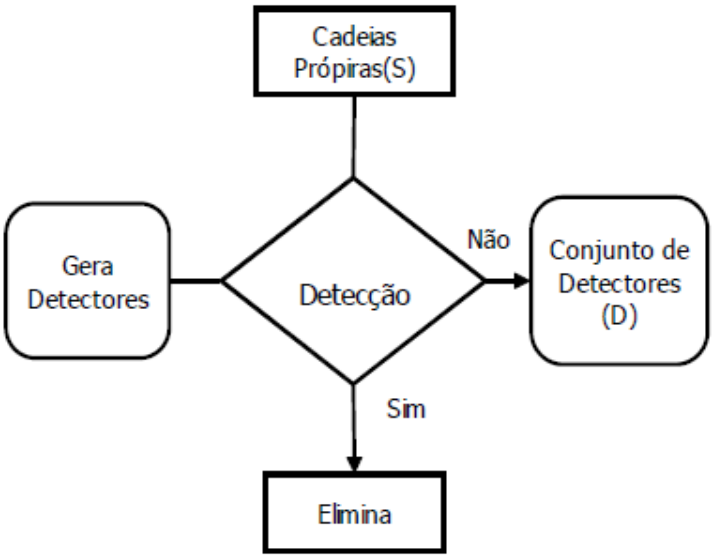

Figura 8 - Fase de Censoriamento - Proposta Original do ASN (AMARAL, 2006, p. 50)

Na fase de Censoriamento, apresentada na Figura 8, são definidas as cadeias próprias, e logo após são gerados detectores, que serão confrontados com as cadeias próprias,

e se a afinidade entre os dois for maior do que o determinado, estes detectores são eliminados, caso contrário são armazenados em um conjunto de detectores. A Figura 9, a seguir, demonstra o monitoramento de anomalias.

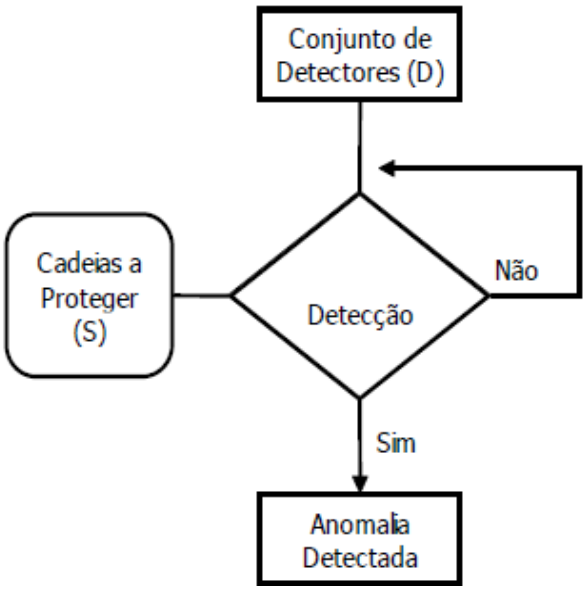

Figura 9 - Fase de Monitoramento - Proposta Original do ASN (AMARAL, 2006, p. 50)

Logo após a geração do conjunto de detectores, estes são aplicados a cadeias que se deseja proteger. Se o nível de afinidade entre eles for menor do que o determinado, então foi encontrado um elemento não-próprio, ou seja, uma anomalia.

Já autores posteriores, como Amaral (2006, p. 18), consideram o algoritmo contendo três fases:

1. definição do que é próprio: definição dos dados próprios, ou seja, aqueles que se deseja proteger, para que possa formar um padrão do que é próprio através desde;
2. geração dos detectores: geração a avaliação dos detectores, com base nos dados próprios;
3. monitoração de ocorrências de anomalias: avaliação feita pelo os detectores em dados que se deseja proteger.

Para a definição dos dados próprios deve-se levar em consideração o que deseja proteger

Para Balachandran (2005, p. 13), o algoritmo da Seleção Negativa possui uma série de características que o difere dos demais:

- não há necessidade de um reconhecimento prévio de um intruso;
- detecção probabilística e ajustável, de acordo com a quantidade de detectores;

- detecção distribuída, ou seja, poderá fazer a detecção apenas em partes do todo que se protege, os detectores são executados de forma independe até que uma alteração é detectada;
- a detecção é local. Pequenos blocos de dados são verificados, quando um detector não encontra uma anomalia, esta poderá ser encontrada na próxima sequência de dados a serem verificadas;
- o conjunto de detectores locais pode ser único, ou seja, se um local é atacado, outro poderá estar protegido por causa do conjunto de detectores serem independentes, o que diminui a probabilidade de falhas em sistemas.
- o conjunto de detectores trabalha de forma mútua, assim um conjunto é capaz de oferecer "proteção" ou "ajuda" a outro conjunto;

Um dos problemas com a utilização do ANS é o custo da geração de detectores em relação ao tamanho do conjunto de cadeias, pois como as cadeias são geradas aleatoriamente, pode acarretar a geração de cadeias repetidas.

Vários autores propuseram melhorias e modificações ao algoritmo original, como a inclusão de outro algoritmo para realizar a seleção dos detectores. Gonzalez *et al* (2002), propôs uma modificação no Algoritmo, chamando-o de RNSA, *Real-Valued Negative Selection Algorithm*, que trabalha com valores reais para a representação de detectores. Suas principais vantagens são: Gonzalez *et al* (2002, *online*)

- resolução do problema inicial, onde, na maioria das vezes, os detectores são mapeados de volta ao estado original;
- é possível a utilização de algoritmos auxiliares, por exemplo, para verificar o elemento mais próximo;
- o algoritmo facilita a aplicação de técnicas de aprendizagem de máquina, o que eleva o nível de conhecimento da aplicação, facilitando a extração de informação útil.

Para o desenvolvimento do algoritmo é necessário conhecer o domínio ao qual deseja aplicar o mesmo, pois assim como cada Sistema Imune Natural é único a implementação de tal algoritmo também será único, pois será especifico para resolver um determinado problema. Deve-se ter em mente o que deseja proteger e trabalhar os detectores em cima de dados próprios para criar um padrão de detecção.

A próxima seção apresenta alguns trabalhos relacionados a Sistemas Imunológicos Artificiais para resolução de problemas.

2.5. Trabalhos Correlatos

Muitos trabalhos já foram produzidos aplicando a técnica de Sistemas Imunológicos Artificiais à detecção de *spams.* Por exemplo:

- Guzella *et. all* (2006) relatam em um artigo a experiência que obtiveram com a modelagem de um SIA para a detecção de *spams*. Com o título: "Modelagem de um Sistema Imune Artificial para a identificação de *SPAM*", os autores abordam as características do e-mail, de forma estruturada, bem como o *spam* e as suas similaridades com o Sistema Imunológico Humano e a partir dessas similaridades e do algoritmo de seleção clonal (CLONALG), desenvolveram um modelo para a detecção de *spams*, onde o mesmo é confrontado com o filtro bayesiano, obtendo um resultado melhor em relação a detecção de *spams*, porém com menor taxa de acerto em relação aos e-mails legítimos e ainda conta com um tempo de processamento maior. Em 2008, os mesmo autores desenvolveram o artigo: "Identificação de mensagens de *SPAM* usando uma abordagem inspirada no sistema imunológico", no qual propuseram a detecção de *spams* através do IA-AIS - Sistema Imunológico Artificial inato e adaptativo, o qual foi comparado com filtro bayesiano, obtendo um melhor resultado em relação a análise de e-mails válidos detectados como *spam* (falso positivo). O desenvolvimento do modelo proposto se dá através da junção de características da Seleção Negativa (reconhecimento do próprio e não-próprio) e a Seleção Clonal (seleção e reprodução de células).
- Oda e White (2004) abordam a evolução do e-mail e do *spam* e como um anti-*spam* pode ser adaptar a essas mudanças. Um aspecto interessante abordado sobre as mudanças é com relação aos novos contatos do usuário e novos idiomas que os mesmos podem apresentar, um exemplo poderia ser um usuário brasileiro que passou a ter contato com usuários do Japão, como um anti-*spam* trataria esse novo padrão, até então desconhecido, ou seja, não próprio, como um próprio, assim os autores propõem um sistema com a capacidade de lembrar e também de esquecer. No artigo, o *spam* é tratado como um agente patogênico (causador de doenças), o e-mail como antígeno e as informações digitais são os linfócitos, no qual cada um compõe um padrão para o anticorpo. O trabalho se desenvolve em torno da criação e eliminação de linfócitos, no qual são treinados a detectar um *spam* e um e-mail legítimo, assim quando algum padrão de e-mail que antes era detectado como *spam*,

deixa de ser *spam*, este é substituído por outro por um período de tempo, esta substituição caracteriza a capacidade de esquecer.

- Silva (2009) desenvolveu sua dissertação de mestrado sobre a segurança de rede de computadores, mais precisamente tentando descobrir intrusos na rede. Com o objetivo de detectar intrusos que utilizam *ping scan*, uma ferramenta utilizada por administradores de rede para obter informações, que se usado por pessoas maliciosas poderá obter dados referentes à rede. Para resolver essa problemática, o autor propõe uma abordagem baseada na teoria do perigo, que segundo Silva (2009, p. 14) é um novo ramo de pesquisa sobre Sistemas Imuno-inspirados. A teoria do perigo diz que o sistema imune realizará defesa enquanto houver uma situação de perigo (SILVA, 2009, p. 15), contendo uma vantagem em relação a Teoria da Seleção Negativa, a redução de falsos alertas, pois não requer um padrão normal para que seja detectado o perigo real. A implementação dessa teoria é baseada nas células dendríticas, que deu origem ao DCA - *Dendritic Cells Algorithm* (Algoritmo das Células Dendríticas), que propõe a detecção de anomalias através de sinais que evidenciam o perigo. Os sinais de perigo são classificados de acordo com métricas pré estabelecidas, na qual possuem um valor superior e um inferior, e o que estiver dentro destes valores será visto dentro da normalidade. O teste realizado pelo autor pode comprovar a eficiência do algoritmo para a detecção de intrusos na rede provenientes da utilização da ferramenta *ping scan*.

Além dos trabalhos citados acima há também vários outros que utilizam técnicas evolutivas na detecção de *spams*. O Sistema Imunológico Artificial já se mostrou eficiente em várias pesquisas, sendo esta uma técnica promissora para agregar a detecção de *spam*. A seção a seguir apresenta os materiais utilizados para o desenvolvimento deste trabalho bem como a metodologia abordada.

3 MATERIAIS E MÉTODOS

Esta seção apresenta os recursos utilizados para a elaboração deste trabalho, como local de realização, materiais utilizados e a metodologia de trabalho. Tais recursos juntamente com a orientação obtida, permitiram o desenvolvimento e a conclusão do trabalho.

3.1. Local e Período

O desenvolvimento deste trabalho ocorreu durante o 2º semestre de 2011, como requisito parcial da Disciplina de Trabalho de Conclusão de Curso I e no 1º semestre de 2012, como requisito parcial da Disciplina Trabalho de Conclusão de Curso II, nos laboratórios de informática do Centro Universitário Luterano de Palmas e na Residência própria.

3.2. Materiais

Para que fosse possível a realização deste trabalho foram utilizados recursos disponibilizados pelo CEULP/ULBRA e recursos próprios. O recurso disponibilizado pela Instituição foi o ambiente físico adequando para a realização do trabalho. O hardware utilizado para o desenvolvimento do trabalho foi um Core 2 Duo, 2.4GHz e 3 GB de memória RAM (recurso próprio).

Também foram utilizados para o desenvolvimento do trabalho, sites contendo informações sobre spams, como portais de *markenting*, sites de hospedagens e sites sobre contenção de *spams*. Estes sites serviram como base para compor o banco de palavras de *spams*, ou seja, as palavras e/ou termos mais comuns em *spams*.

Outras bases utilizadas foram (http://www.em.ca/~bruceg/spam/) que é um diretório que armazena *spams* recebidos desde 1998 e também a base de dados do SpamAssassin disponivel em: (http://spamassassin.apache.org/publiccorpus), esta armazena *spams* e emails válidos,

3.2.1. Software

Para o desenvolvimento da aplicação foi utilizada a linguagem de desenvolvimento Java, a IDE (*Integrated Development Environment*) escolhida foi NetBeans versão 7.1.1. NetBeans é

um ambiente integrado para desenvolvimento *free*, cujo o download pode ser obtido em: (http://netbeans.org/downloads/7.1.1/).

3.2.2. Fontes Bibliográficas

Entre os materiais utilizados para o referencial teórico foram retirados de bibliotecas on-line, como o Google *Books* e os mais diversos materiais on-line. Os tipos de materiais foram:

- artigos científicos;
- dissertações de mestrado;
- teses de doutorado;
- livros, inclusive on-line;
- sites que abordam o assunto;

Tais referências podem ser visualizadas na seção 6, de Referências. E ainda, para uma melhor visualização do conteúdo teórico e entendimento do mesmo, foram visualizados vídeos, em sites como *youtube* e *vimeo*, que demonstram o funcionamento das teorias abordadas.

3.3. Metodologia

O presente trabalho constitui em um estudo sobre o Sistema Imunológico Artificial e a aplicação desde para a detecção de *spams*. Para tanto, foi necessário um entendimento sobre o Sistema Imunológico Humano, no qual foram estudados seus conceitos e estruturas. Posteriormente, partiu-se para o estudo do Sistema Imune Artificial, no qual se estudou os três principais modelos: Teoria da Seleção Clonal, Teoria da Rede Imunológica e Seleção Negativa. Tais estudos serviram de base para a escolha do algoritmo que compôs o resultado do trabalho. Na sequência, foi realizado um estudo sobre o domínio ao qual o algoritmo seria empregado. Para o domínio escolhido (detecção de *spams*) foram necessários estudos sobre e-mails e *spams*. No caso, foi dado uma importância maior a *spams*, pois este seria o dado a ser detectado. Foram analisadas estatísticas sobre *spams*, categorias de *spams* e técnicas de detecção.

Após os estudos que permitiram a compreensão do tema do trabalho, foi definido o algoritmo a ser utilizado, no caso, o Algoritmo de Seleção Negativa. Este foi escolhido por possibilitar a distinção do próprio e não-próprio. Dentro do contexto deste trabalho, o próprio equivale a um e-mail válido (não *spam*) e o não-próprio equivale a um *spam*.

Após a escolha do algoritmo para a detecção de *spams*, deu-se início a modelagem do mesmo, escolha da linguagem, métodos de desenvolvimento e restrições. O desenvolvimento da aplicação consistiu em três fases: geração dos detectores, avaliação dos detectores e monitoramento. Na fase de geração dos detectores, os mesmos são gerados aleatoriamente a partir de dados não próprio, no qual cada detector é formado por uma cadeia de seis termos. Posteriormente, os detectores gerados são avaliados. A geração e avaliação dos detectores fazem parte da primeira etapa do algoritmo, que é o censoriamento.

Para tanto, foram escolhidos os dados próprios para a etapa de avaliação dos detectores. Estes dados consistem em e-mails válidos, ou seja, não *spams.* É importante ressaltar que estes dados (e-mails) foram retirados da caixa de entrada de uma conta de e-mail e armazenados em um arquivo texto, juntamente com e-mails retirados de uma base de dados web. Depois de gerados os detectores, os mesmos foram avaliados. Para fazer a avalição, foi desenvolvido um método que permite medir a similaridade entre textos, o qual foi baseado na distância de Hamming, que avalia a distância entre duas cadeias de caracteres. O resultado desta distância indica se o conjunto de detectores gerado irá detectar um dado próprio (e-mail válido) como não próprio (*spam*), o que acarreta em um falso positivo (*e-mail* válido detectado como *spam*). Se o resultado for positivo, ou seja, se este conjunto de detectores identificou um e-mail como *spam*, logo este detector não será armazenado (ou seja, serão descartados). Esta fase de avaliação é importante para que futuramente um detector não detecte um e-mail válido como *spam*. Esta fase também contempla o reconhecimento de padrões, em que os detectores são treinados a não reconhecer um dado próprio. Depois de avaliados, os detectores que passaram pela seleção foram salvos em um arquivo texto.

Por fim, a última etapa consiste na fase de monitoramento, no qual os detectores que passaram pela Seleção Negativa avaliam os dados que se deseja proteger. No caso, esta avaliação também utiliza a distância de Hamming. Na monitoração foram utilizados arquivos determinados como *spams* e arquivos determinados como e-mails válidos, no qual o algoritmo será capaz de identificar *spams*. Antes de fazer tal avaliação, todos os arquivos passam por uma fase de pré-processamento, que inclui a eliminação de *stopwords*, remoção de tags HTML e normalização das palavras. Na análise, todos os detectores avaliam um arquivo por vez, no qual é aplicada a distância de Hamming para medir a similaridade entre as palavras do arquivo e as palavras dos detectores. A cada palavra similar encontrada, foi incrementada uma variável para verificar a quantidade de palavras semelhante entre um detector e o arquivo analisado. Logo depois é contada a quantidade de detectores que tiveram mais da metade de

palavras semelhantes com o arquivo analisado. Ao final, se a maioria dos detectores julgou o arquivo como *spam*, este deverá ser enquadrado como tal.

Por fim, serão apresentados os resultados no que tange a falsos positivos e falsos negativos e, a partir destes, serão calculadas medidas de desempenho através das métricas de precisão e *recall*, e ainda a comparação destes resultados com resultados de análise de ferramentas baseadas em redes bayesianas. Após as análises, serão apresentadas as considerações finais do trabalho, bem como trabalhos futuros.

4 RESULTADOS E DISCUSSÃO

Esta seção apresenta o desenvolvimento de uma aplicação para a detecção de *spams* baseada no conceito de Sistema Imunológico Artificial. Para isto, o primeiro passo foi a definição do algoritmo para a detecção de *spam*. Nesta etapa, foi escolhido o Algoritmo de Seleção Negativa, por propiciar a distinção do próprio e não-próprio. A partir desta escolha foram definidos outros passos, como definição do próprio, o método para geração de detectores, a escolha do método para medir a similaridade e determinar o limiar de similaridade. O limiar vai determinar o ponto de corte dos detectores durante a avaliação e para determinar classificar os e-mails na fase de sensoriamento. Estes passos serão descritos nas subseções seguintes. A próxima seção apresenta o modelo proposto para o desenvolvimento da aplicação baseada no algoritmo de Seleção Negativa.

4.1. Algoritmo Proposto

A aplicação desenvolvida é baseada no algoritmo de Seleção Negativa. Este algoritmo é dividido em duas fases: o censoriamento, sendo que esta é composta pela definição dos dados próprios, geração dos detectores e avaliação destes; e o monitoramento, que contém a avaliação dos dados que se deseja proteger ou avaliar.

Os dados ou cadeias próprias são aqueles que se deseja proteger, ou seja, para esta aplicação são os e-mails válidos. Na primeira etapa os e-mails válidos são utilizados para selecionar os detectores, assim os detectores que forem muito próximo do dado analisado, ou seja, semelhante serão eliminados, pois futuramente, na fase de censoriamento, tais detectores poderiam identificar um e-mail válido como *spam*.

No caso, os detectores são formados por palavras que são mais comuns em *spams*. Esta lista de palavras que formam os detectores é composta por palavras e/ou termos, retirados de sites de *markenting* como (http://www.salesnexus.com), sites de hospedagens como (http://www.activewebhosting.com) e sites sobre contenção de *spams*. Cada detector é formado por um conjunto de seis palavras/termos selecionadas aleatoriamente desta referida lista.

No algoritmo proposto originalmente por Forrest em 1994 é gerado um conjunto de detectores e estes detectores são submetidos a cadeias próprias. Para que estes detectores sejam selecionados, é utilizada uma medida de similaridade, no qual calcula o quão similar é

um detector e os dados próprios. Se a similaridade for superior à definida através de um limiar, o detector é descartado, pois ele estar detectando algo próprio, neste caso um e-mail válido.

Tal reconhecimento se faz necessário para que futuramente tal detector não faça um reconhecimento de em um e-mail válido, assim na fase de avaliação os detectores que reconheceram um dado próprio são eliminados, pois nesta fase de seleção são selecionados aqueles que não fizeram tal reconhecimento.

Os *spams*, na maioria das vezes, possuem características em comum, como o cabeçalho duvidoso, ou seja, e-mails cujo cabeçalho contém remetente desconhecido, e também o conteúdo da mensagem. Este se diferencia pela forma de disponibilização de seu conteúdo, ou seja, a mensagem pode vir com caracteres trocados, pode estar em outra língua, ou até mesmo ser apresentada por meio de uma imagem.

De posse dessas informações, a geração de detectores será feita a partir de palavras que são mais comuns em *spams*, obtidas de sites contendo informações sobre spams, como portais de *markenting*, sites de hospedagens e sites sobre contenção de *spams*. Em seguida, os detectores gerados serão avaliados, como já foi dito, a partir de uma base de dados de e-mails válidos (cadeias próprias).

Assim, estes detectores serão eliminados caso façam o reconhecimento de um e-mail legítimo como *spam* ou mantidos, caso não reconheçam o e-mail como *spam.* A Figura 10 apresenta os passos que compõem a metodologia de desenvolvimento da aplicação.

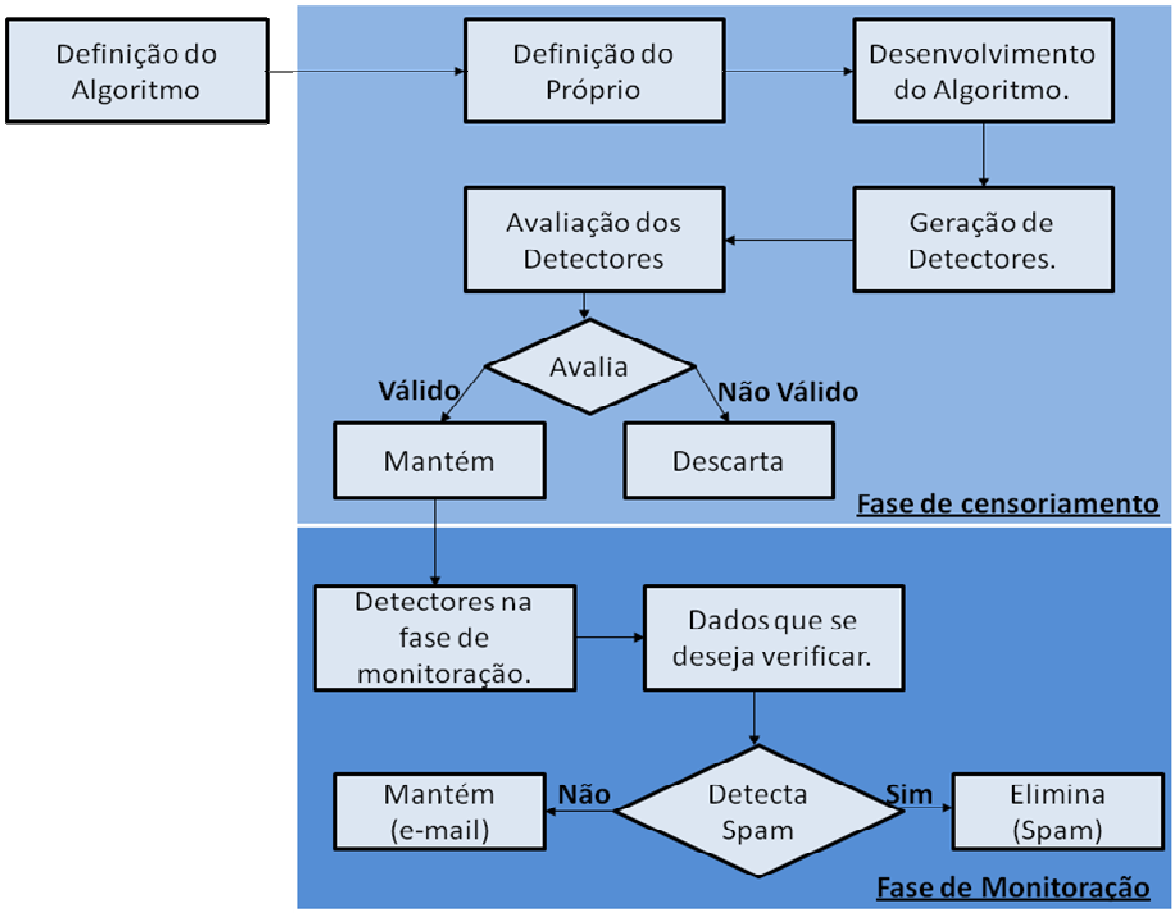

Figura 10 – Metodologia de Desenvolvimento

A Figura 10 apresenta todo o processo de desenvolvimento da aplicação, que vai desde a definição do algoritmo até a fase de monitoração que é a etapa final da aplicação. Após as definições da aplicação, como escolha do algoritmo e definição dos dados próprios, é dado início ao desenvolvimento.

O desenvolvimento da aplicação começa com a geração de detectores. Tais detectores são gerados de forma aleatória e então estes passam por uma avaliação. A partir desta avaliação são mantidos, caso não façam o reconhecimento dos dados próprios (ou seja, e-mails válidos), ou descartados, caso reconheçam estes mesmos dados próprios.

Após a fase de censoriamento, os detectores que passaram pela seleção vão para a fase de monitoração dos e-mails, no qual os detectores selecionados os avaliam (*spams* ou não). Durante a avaliação, os detectores são comparados com os e-mails e, se a semelhança entre detectores e o e-mail avaliado for "próxima" este e-mail será classificado como *spam*. A medida de similaridade utilizada para verificar a semelhança, foi a medida de Hamming. A Figura 11 representa o funcionamento da aplicação em um ambiente real.

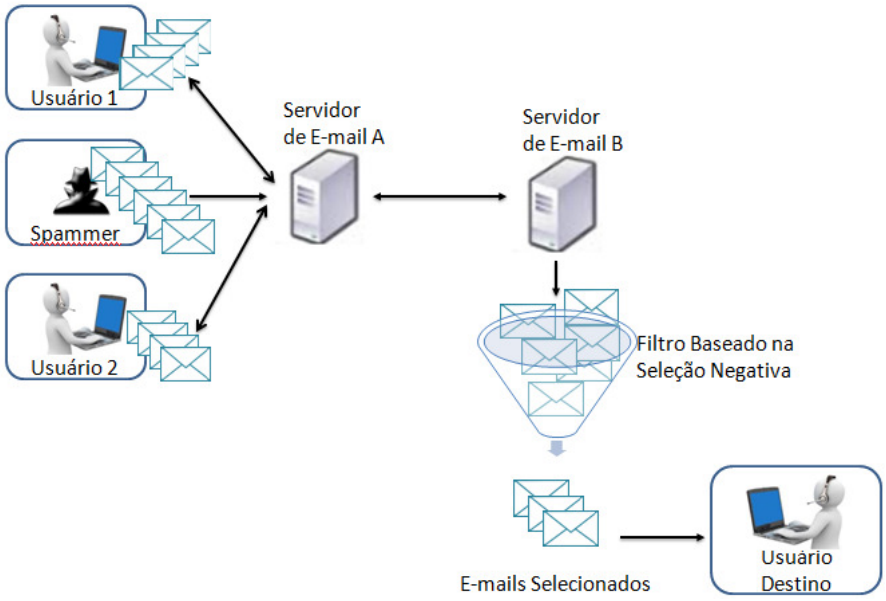

Figura 11 – Demonstração da Aplicação em um Servidor de e-mail

A Figura 11 representa a aplicação desenvolvida, no qual contempla somente a fase de monitoramento, no qual os detectores já devem ter passado pelo reconhecimento de padrão. Neste exemplo, usuários, incluindo *spammers*, do servidor de e-mail A enviam e-mails, dentre eles *spams* e e-mails válidos, para um usuário do servidor de e-mail B. Ao receber os e-mails, o servidor de e-mails B faz a seleção destes, através da aplicação, enviando para a caixa de entrada do usuário apenas e-mails válidos (não *spams*). Vale lembrar que esta aplicação seria apenas um complemento para os filtros já existentes, os e-mails continuariam passando pelos filtros estáticos e que este é uma visão da aplicação final, neste trabalho, será apenas simulado um ambiente para testes em que a aplicação será capaz de classificar e-mails.

A seção seguinte apresenta os passos para o desenvolvimento da aplicação.

4.2. Desenvolvimento da Aplicação

Após a escolha do algoritmo de Seleção Negativa, o desenvolvimento deste ocorreu utilizando a linguagem de desenvolvimento Java, onde foram utilizados métodos Java já existentes como os métodos `toLowerCase()` `replaceAll()`, os quais foram necessários para a normalização dos e-mails, além do desenvolvimento de outros métodos específicos para este trabalho. A aplicação é composta por quatro classes: a classe principal; a classe de pré-processamento de e-mails; a classe de censoriamento, na qual são gerados e classificados os detectores; e a classe de monitoração que é destinada a fazer a distinção do próprio (e-mails válidos) e não próprio

(*spams*). A classe de Censoriamento.java é composta pelos métodos apresentados na Tabela 2.

Tabela 2 - Métodos da Classe Censoriamento

Métodos	Descrição
geraDetectores()	Classe responsável por gerar cadeias de detectores, os quais são gerados de forma aleatória.
avaliaDetectores(ArrayList detectores)	Recebe como parâmetro um ArrayList contendo os detectores. Cada posição do vetor contém um conjunto de detectores, que serão avaliados de acordo com o grau de similaridade entre os detectores e os dados próprios. Esta similaridade é medida através do método de hamming().
armazenaDetectores(String detectoresSalvar)	Passado pelo método de avaliação, os detectores são armazenados. O método armazenaDetectores() recebe como parâmetro uma String de detectores e estas são armazenadas em um arquivo texto, no qual cada linha contém um conjunto de detectores.
hamming(String a, String b)	Este método recebe como parâmetro duas Strings a serem analisadas e tem como retorno o grau de similaridade entre as *Strings* passadas.

Os métodos apresentados na Tabela 2 serão mais detalhados nas próximas subseções. A Tabela 3 apresenta os métodos utilizados na classe de 'Monitoração.java'.

Tabela 3 – Métodos da Classe de Monitoração

Métodos	Descrição
verificaEmail()	Método responsável por verificar se o e-mail é um *spam* ou não. Este método conta com o método verificaTermos().
verificaTermos(String a String b)	Verifica se o detector que irá analisar o arquivo possui algum termo, se sim verifica se no arquivo há um termo semelhante, neste método também é utilizada a distância de Hamming.

O método hamming(), apresentado na Tabela 2, o qual foi utilizado para avaliação do detectores, ou seja, medir a similaridade destes com os dados próprio, também é utilizado na classe de monitoração para medir a similaridade entre os detectores e o e-mail analisado. Este método será melhor detalhado posteriormente.

É necessário que os e-mails utilizados na seleção dos detectores e os e-mails que passarão pela análise passem por uma fase de pré-processamento, na qual será feita uma "limpeza" no texto dos arquivos. Para tanto, a aplicação conta com uma classe para efetuar o pré-processamento dos e-mails. Esta classe é responsável por eliminar os *stopwords*, eliminar cabeçalho e normalizar o texto. A classe chama-se: LimpaEmail.java e é composta pelos seguintes métodos:

- método *String* limpaCabecalho(e-mail): recebe como parâmetro uma *string* contendo o texto de e-mail completo e retorna o e-mail sem o cabeçalho e assunto, retornando apenas o corpo do e-mail, que é a parte que passará pela análise;
- método *String* getLimpaEmail(e-mail): este método é responsável por eliminar os *stopwords* presentes no e-mail, logo após o e-mail passa por uma normalização;
- método *String* normaliza(e-mail): também recebe uma *String* contendo um e-mail e retorna esta *String* normalizada, ou seja, retira todos os acentos, *tags* HTML, pontuações e passa todo o texto para minúsculo. Para o desenvolvimento do método normaliza() foram utilizados os métodos java replaceAll(), o qual trabalha com expressões regulares e foi utilizado para retirar as *tags* HTML, acentos e pontuações e o método toLowerCase(), que foi utilizado para converter *String* em minúsculo.

A técnica utilizada no método getLimpaEmail() foi a de *Stopwords*, que tem como objetivo eliminar dados irrelevantes no texto, ou seja, aqueles que aparecem com muita frequência, como artigos, preposições e conjunções e, para este caso de análise de e-mails, inclui também a lista de saudações, pois são muito comuns nos e-mails. Estes dados eliminados não têm implicação no resultado final da análise além de diminuir o número de comparações desnecessárias. A Tabela 4 apresenta um exemplo da aplicação da técnica de *Stopwords.*

Tabela 4 - Exemplo da remoção de *Stopwords*

Conteúdo de E-mail	Lista de *Stopwords*	Termos Relevantes
Olá, **segue** em **anexo** o **trabalho**. **Estude** para a **prova**, pois **será difícil** de **passar**.	Olá	segue
	em	anexo
	o	trabalho
		Estude
	a	prova
		será
	de	difícil
		passar

No exemplo ilustrado na Tabela 4, no qual é aplicada a técnica de *Stopwords*, é possível observar que os elementos retirados do texto não são relevantes para a verificação ou não de *spams*. Isto porque tais elementos fazem parte da lista de *Stopwords,* que são palavras que não formam um termo para a análise de *spams*, como "click here" ou "Clique aqui". Sendo assim, palavras como aqui e "here" não entrarão na lista de *stopwords*, pois a retirada destas palavras poderia interferir no resultado final da análise dos termos.

A seção seguinte apresenta a primeira etapa do desenvolvimento da aplicação, o desenvolvimento da classe de Censoriamento, que contempla os métodos de geração e avaliação dos detectores.

4.2.1. Classe de Censoriamento

A aplicação desenvolvida tem início com a fase de censoriamento na qual são definidos os dados próprios, neste caso os e-mails válidos, que não são *spams*. Depois são gerados os detectores e avaliados a partir dos dados próprios. A geração dos detectores foi feita a partir de um arquivo de texto contendo palavras e/ou termos que mais aparecem em *spams*, totalizando 1300 palavras/termos. Um termo é formado por uma palavra ou um conjunto de palavras, no qual representa um conceito.

Para esta aplicação foram utilizados no máximo dois termos, pois a estrutura de dados utilizada não é apropriada para trabalhar grande volume de dados, a utilização de outra estrutura possibilitaria um gerenciamento melhor dos dados além de otimização nas operações. O arquivo foi montado a partir de bancos de palavras mais comuns em *spam*,

retirados de sites de publicidade, como (http://www.salesnexus.com) e sites de hospedagem de domínio, como (http://www.activewebhosting.com).

A próxima seção apresenta o método para a geração de Detectores.

4.2.1.1. Método de geração de Detectores

Para a formação dos detectores, foram utilizados termos mais comuns em *spams*. Tais termos foram organizados em um arquivo texto, de modo que cada palavra ou termo fique em uma linha do arquivo. Este arquivo foi denominado "BancoPalavrasSpams.txt".

Para o desenvolvimento da geração de detectores foi criado o método geraDetectores(). Este método faz a leitura do arquivo que contém as palavras/termos e armazena cada termo em uma posição de um *array*. Para a criação do conjunto de detectores, foi criado um *ArrayList*, no qual o mesmo irá receber, em cada posição, seis posições do *array*, ou seja em uma posição do *ArrayList* terá seis posições do *array*, tal atribuição será de forma aleatória através do método random() do Java, o qual gera índices aleatórios. A Figura 12 apresenta parte dos detectores gerados.

```
[isn't junk|passwords|renda extra|tripple x|apply online|viagra and]
[earn|vacation offers|the dollar|free offer|asthma|registered with]
[in lifetime|big bucks|reverses aging|instant access|zolus|percent guaranteed]
[despachamos para|eliminate|remove me|bargain|in accordance|free consultation]
[off shore|formulário|save up|copy dvds|creditors|espionagem]
```

Figura 12 – Parte do conjunto de detectores

Cada termo de um detector foi separado pelo caractere "|". A seção seguinte apresenta o método utilizado para medir a similaridade. Este método foi utilizado para medir a similaridade entre os detectores a serem avaliados e os dados próprios e também a similaridade entre os detectores e os dados que se deseja avaliar.

4.2.1.1. Método para Medir a Similaridade

Existem diversas formas para fazer a comparação entre duas cadeias de caracteres, ou seja, verificar o quanto estas cadeias são semelhantes. Algumas destas formas são: distância de Hamming[4], distância de Levenshtein[5], distância Euclidiana[6] e medida do cosseno[7]. A forma

[4] Método utilizado para verificar a similaridade entre duas cadeias de igual tamanho.
[5] "quantidade mínima de operações necessárias para transformar uma cadeia de caracteres em outra qualquer" (DRAGO, 2007, p. 22).
[6] Comparação entre duas séries temporais.

utilizada para este trabalho para medir a similaridade foi a distância de Hamming, que segundo Beuren (2010, p. 27), também vem sendo utilizada como medida de distância entre sequências de DNA, além de ser amplamente utilizada na área de telecomunicações para a detecção de erros em transmissões de dados.

A distância de Hamming se faz necessário para medir a similaridade, na primeira fase do desenvolvimento é aplicado na avaliação dos detectores, no qual mede a semelhança entre os detectores gerados e os dados próprios. Na segunda etapa é utilizado para monitoração dos dados que se deseja verificar, ou seja, a detecção de *spams*.

Para obter a distância de Hamming são dados dois vetores de caracteres, X e Y, de forma que a distância é calculada como mostra a Figura 13.

$$D = \sum_{i=1}^{L} \delta(X_i , Y_i)$$

Figura 13 - Forma de Hamming

Onde D representa o resultado do somatório da função dada por δ (Xi,Yi) que será igual a 1 caso (Xi ≠ Yi) e igual a 0 caso contrário, assim δ representa o número de posições onde Xi e Yi diferem-se.

Para o desenvolvimento de Hamming, foi criado o método hamming(), que receberá duas *strings* do mesmo tamanho como parâmetro e terá como retorno o número de caracteres diferentes entre as palavras. Por exemplo: a palavra “viagra”, que é amplamente utilizada em *spams* sendo escrita de diferentes maneiras como “vi4gr4”. Neste caso, a distância entre a palavra “viagra” e “vi4gr4” é igual a 2, logo, as duas palavras são consideradas “próximas” por ter uma distância pequena uma da outra. Usando outros métodos de comparação presentes no Java, como o equals(), não se chegaria ao mesmo resultado, pois o resultado da comparação pelo equals() é atômica. Figura 14 ilustra o processo de comparação por Hamming.

v	i	a	g	r	a
v	i	4	g	r	4
0	0	1	0	0	1

Figura 14 - Comparação por Hamming

[7] Cálculo do cosseno do entre dois vetores.

Observando a Figura 15, é possível verificar que é feita uma comparação, caractere a caractere, entre as duas cadeias (X e Y). Se os caracteres contidos nas mesmas posições forem iguais, o valor do retorno é 0 (zero) e se forem diferentes, o retorno será 1. Após a comparação é feito o somatório dos campos cujo resultado foi 1. Logo, para este exemplo, a distância de Hamming entres as cadeias comparadas é 2, pois dois campos diferem-se. A Figura 15 apresenta o método desenvolvido para calcular a distância por Hamming.

```
public static int hamming(String a, String b) {
    int d=0;
    for (int i=0; i<a.length(); i++){
        if (a.charAt(i) != b.charAt(i))
            d=d+1;
        else
            d=d+0;
    }
    return d;
}//fim do metodo hamming
```

Figura 15 - Método para Medida de Similaridade

O método apresentado na Figura 15 representa o cálculo da medida de similaridade por Hamming. Este mesmo método foi utilizado para a seleção de detectores e também para a monitoração de e-mails. Como dito, o método recebe duas S*tring*s de igual tamanho como parâmetro e retorna 0 (zero) caso todas as posições das duas *strings* forem iguais, ou diferente de zero caso as *Strings* tenham valores diferentes na mesma posição sendo que para cada posição diferente é incrementado 1 no valor da distância (d). Este, porém, não é o resultado final, sendo aplicado apenas para medir a similaridade entre as palavras. A seção seguinte apresenta o método de avaliação dos detectores gerados.

4.2.1.2. Método de Avaliação de detectores

Os dados com os quais o conjunto de detectores foi avaliado são de uma conta de e-mail pessoal. Esta base, no formato texto, é composta por 159 e-mails válidos, ou seja, não *spams* e também foram adicionados 50 e-mails válidos retirados da base de dados do SpamAssassin (http://spamassassin.apache.org/publiccorpus), que é um domínio responsável por reunir informações sobre o software Anti*Spam* SpamAssassin.

Após a aplicação do método de medição de similaridade, pelo método de Hamming, a grande maioria dos detectores foi mantida, principalmente porque foram gerados a partir de dados não próprios, ou seja, termos *spams*. Após a verificação, o conjunto de detectores válidos foi salvo em um arquivo de texto, em que cada linha do arquivo contém um detector.

Este arquivo será utilizado na próxima etapa, que é a de monitoração. A Figura 16 apresenta um modelo geral para avaliação dos detectores.

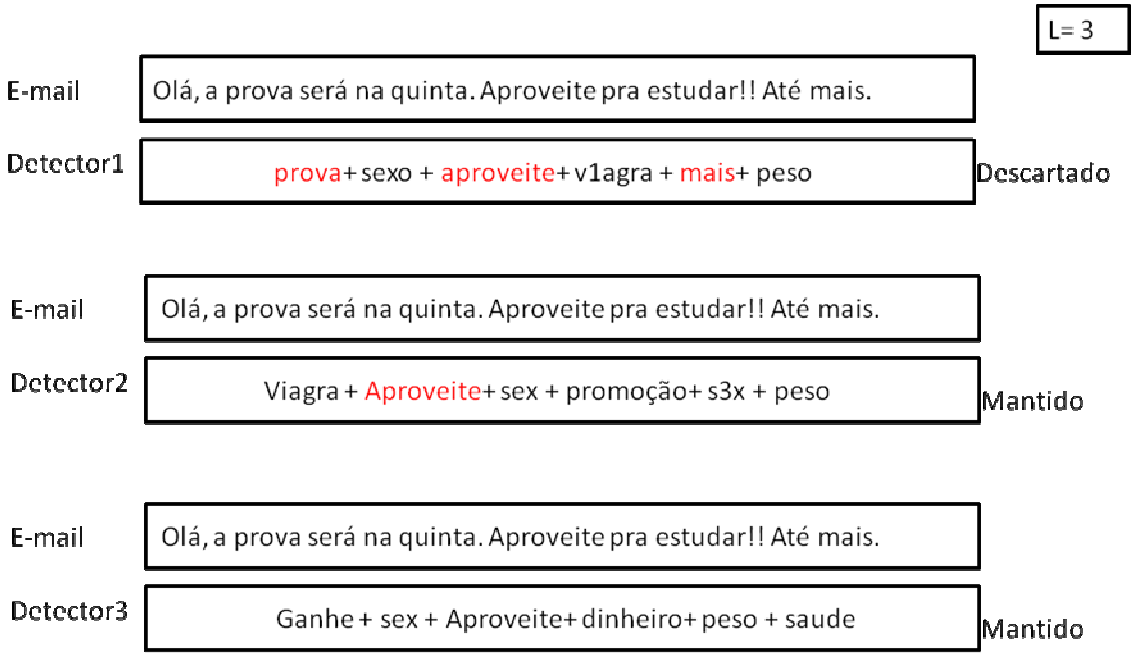

Figura 16 - Modelo geral para regra de avaliação de detectores

O exemplo mostrado na Figura 16 contempla um modelo geral para a avaliação dos detectores. Dado um texto de e-mail legítimo (não *spam*), este é confrontado com um conjunto de detectores. O primeiro detector, o Detector1, formado pelas palavras (prova – sexo – aproveite – v1agra – mais - peso), ao analisar o e-mail, é descartado, pois chegou ao limiar determinado para o descarte, neste caso chegou a 3, logo ele se torna inapropriado, pois futuramente detectaria um dado próprio o que não ocorre com o Detector2 e Detector3 que são mantidos.

Este modelo é apenas uma explanação geral para o entendimento do algoritmo. Antes de chegar a este modelo, ainda passa-se pelo método de similaridade entre as palavras e então, se a metade ou mais das palavras de um detector for similar ao dado próprio, este detector é descartado.

A seção a seguir apresenta a classe de monitoração, na qual foi desenvolvido o método pra monitorar as ocorrências de *spams*.

4.2.2. Classe de monitoramento

A classe de monitoramento consiste na segunda fase da aplicação, que é a verificação dos dados que se deseja proteger, neste caso os e-mails válidos. Para tanto, foi criado um método do tipo void chamado verificaEmail(), sem parâmetro, pois, todos os arquivos de e-mails a

serem analisados serão carregados dentro do código e serão analisados de uma só vez. Porém, este código poderia ser reescrito recebendo como parâmetro o arquivo que se deseja analisar e tendo como retorno uma variável do tipo booleana no qual retornaria *true* caso identificasse um spam ou *false* caso contrário. A nova forma de implementação deste método seria interessante caso o mesmo venha a ser aplicado em um servidor de e-mail.

O modelo desenvolvido para esta aplicação irá verificar todos os arquivos de uma vez e avaliar o quais são *spams* e quais não são. A Figura 17, a seguir, apresenta o processo de verificação de e-mails.

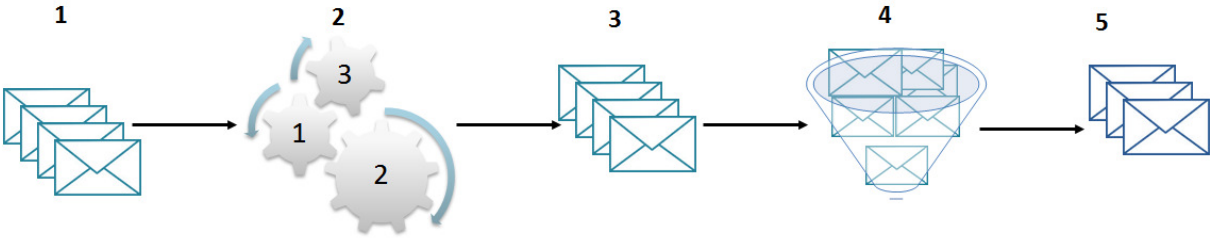

Figura 17 – Processo de verificação de e-mails

A Figura 17 representa cinco etapas: a primeira apresenta os dados que se deseja verificar; a segunda mostra o pré-processamento dos arquivos, o que inclui a eliminação de *stopwords*, a retirada do cabeçalho e das *tags* HTML e a normalização do texto, ou seja, transformá-lo em minúsculos, sem acentos e sem pontuação; a terceira etapa apresenta os arquivos já prontos para passar pela análise; a quarta etapa compreende a análise, na qual inclui o método de Hamming para verificar a similaridade; e, por fim, a última que apresenta os e-mails selecionados.

Na verificação dos e-mails, cada e-mail é verificado por todos os detectores, como pode ser observado na Figura 18, que mostra a interação entre os detectores e os e-mails.

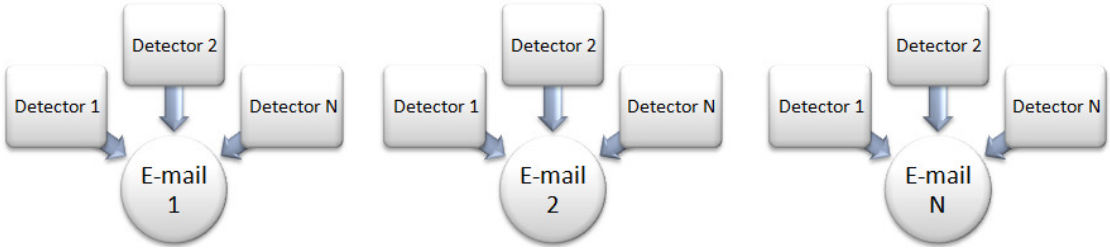

Figura 18 - interação entre os detectores e os e-mails

A Figura 18 apresenta a interação entre os detectores e os e-mails analisados, em que todos os detectores verificam um e-mail. Durante a verificação é chamado o método verificaEmail(), que contém o método verificaTermos() e o método de similaridade; hamming(), que verifica a similaridade entre as palavras/termos dos detectores e as palavras/termos do e-mail, que são do mesmo tamanho. Ao final da verificação o e-mail este é classificado como *spam* ou não. Este processo é repetido até que todos os e-mails sejam verificados.

O pseudocódigo do método de verificação de e-mails será apresentado na Figura 19.

```
1
2   public static void verificaEmail (){
3       inicializa detectores salvos;
4       inicializa arquivos a serem analisados;
5       for (arquivos){
6           for (detectores){
7               contaTermos=contaTermos+verificaTermos(detectores, arquivos)
8               for(detectores.detector){//percorre um detector
9                   for(arquivos.emailProcessado){//percorre um arquivo
10                      if (detectores.detector[i].length() == emailProcessado.arquivo[j].length()){
11                          retorno = hamming(detectores.detector[i],emailProcessado.arquivo[j]);
12                              if(emailProcessado.arquivo[j].length()<=3){
13                                  if(retorno <=1)
14                                      contPalavraSemelhante++;
15                              }
16                              if(emailProcessado.arquivo[j].length()>=4 && emailProcessado.arquivo[j].length()<=5){
17                                  if(retorno <=2)
18                                      contPalavraSemelhante++;
19                              }
20                              if(emailProcessado.arquivo[j].length()>=6 && emailProcessado.arquivo[j].length()<=7){
21                                  if(retorno <=3)
22                                      contPalavraSemelhante++;
23                              }
24                              if(emailProcessado.arquivo[j].length()>=8){
25                                  if(retorno <=4)
26                                      contPalavraSemelhante++;
27                              }
28                      }
29                  }
30                  if (contPalavraSemelhante>=1)
31                      ContPalavrasDetectores;
32              }
33              if(ContPalavrasDetectores>=3)
34                  ContDetectores++;
35          }
36          if(ContDetectores>=5 || contaTermos>=4)
37              imprima("Arquivo Spam: "arquivos.arquivo);
38      }
39  }
```

Figura 19 - Pseudocódigo do método verificaEmail()

Como visto na Figura 19, o algoritmo inicia carregando os detectores salvos e os dados que se deseja verificar. Cada arquivo é verificado por todos os detectores e se o número de palavras semelhante entre o detector e o arquivo for superior a 3 é incrementada a quantidade de detectores que notaram este arquivo como *spam*. Entre as linhas 12 e 27 é verificado o tamanho da palavra que passou pelo método de Hamming, no qual a palavra só será semelhante dependendo do seu tamanho e do retorno do método. Ao final, linha 36, é verificada a quantidade de detectores que advertiu tal arquivo como *spam,* juntamente com a

quantidade de detectores que contém termos. A quantidade mínima de detectores apresentada na linha 36 depende da quantidade total de detectores gerados. Foi estabelecido, através de testes, que a quantidade mínima de detectores seria de 1% para detectores sem termos e 0,8% para detectores com termos.

Para se chegar a tais valores, foram realizados testes com várias quantidades de detectores, no qual verificou-se a quantidade mínima de detectores para que façam uma classificação balanceada entre e-mails e *spams*.

A próxima seção apresenta os resultados dos testes efetuados na aplicação, bem como medidas de desempenho para os resultados.

4.3. Resultados dos testes

Geralmente, para a avalição de filtros Anti*Spams* são utilizados quatro resultados prováveis obtidos a partir da classificação dada ao e-mail recebido, os resultados prováveis são:

- *True Positive* (TP): mensagens *spams* classificadas corretamente;
- *False Positive* (FP): mensagens legítimas classificadas como *spams*;
- *False Negative* (FN): mensagens *spams* classificadas como legítimas;
- *True Negative* (TN): mensagens legítimas classificadas corretamente.

Estes resultados são utilizados para medir o desempenho dos dados classificados. Para a análise dos resultados, foram utilizadas duas medidas de desempenho: precisão e *recall*. Tais medidas são comumente utilizadas para medir o desempenho em Sistemas de Recuperação de Informação, em que se avaliam termos relevantes.

Segundo Assis (2006, p.69), "precisão pode ser definida como o número de mensagens de um conjunto de testes corretamente classificadas em uma categoria dividido pelo número total de mensagens classificadas (correta ou não)". O cálculo para medida de precisão de *spams* e mensagens legítimas é dado da seguinte forma, Figura 20:

$$PS = \frac{TP}{TP + FP} \qquad PM = \frac{TN}{TN + FN}$$

20 (A) 20 (B)

Figura 20 - Precisão de Spams e Mensagens Legítimas

Como pode ser observado na Figura 20, são apresentados os cálculos para a precisão de *spams* e precisão de mensagens legítimas, respectivamente. A Figura 20(A) apresenta o

cálculo para Precisão de *Spams* (PS), que é igual ao total de mensagens categorizadas como s*pams* classificadas corretamente (TP) dividida pelo resultado da soma entre o total de mensagens classificadas como *spams* (mensagens *spams* classificadas corretamente (TP)) e as mensagens legítimas classificadas como *spams* (FP).

Na Figura 20(B) é calculada a precisão sobre as mensagens legítimas, logo a Precisão de Mensagens (PM) é igual às mensagens legítimas classificadas corretamente (TN), sobre o total das mensagens classificadas como legítimas, ou seja, mensagens legítimas classificadas corretamente (TN) mais s*pams* classificados como mensagens legítimas (FN).

Já a análise de desempenho utilizando o *recall* ou sensitividade, Assis (2006, p.69) define como "o número de mensagens de um conjunto de testes corretamente classificados em uma categoria dividido pelo número total de mensagens que são realmente da categoria". Ou seja, são calculadas as mensagens classificadas corretamente divididas pelo total de mensagens da categoria. A Figura 21 apresenta, respectivamente, o *recall* de *spams* (RS) e o *recall* de mensagens legítimas (RM):

$$RS = \frac{TP}{TP + FN} \qquad RM = \frac{TN}{TN + FP}$$

21(A) 21(B)

Figura 21 - Recall de Spams e Mensagens Legítimas

Como pode ser observado na Figura 21(A), Recall de *Spams* (RS) é igual aos *spams* classificados corretamente (TP) dividido pelo total de *spams*, que é dado pela soma dos verdadeiros positivos (TP, *spams* classificados corretamente) mais os falsos negativos (FN, *spams* vistos como e-mail válidos).

Já o *recall* de e-mails válidos é dado pelo quantitativo de e-mails classificados corretamente divididos pela quantidade total de e-mails válidos. A expressão para calcular o *recall* de e-mails válidos foi apresentada na Figura 21(B), no qual *recall* das mensagens (RM) é dado pelas mensagens válidas classificadas corretamente, dividida pelo total das mensagens válidas. Esse total é dado pela quantidade de mensagens classificadas corretamente (TN) mais as mensagens válidas classificadas como *spams* (FP).

Os testes iniciais foram realizados com um total de 100 arquivos, 50 e-mails legítimos e 50 *spams*. Os e-mails (*spams* e não *spams*) foram baixados de uma base de dados disponível em (http://spamassassin.apache.org/publiccorpus), mantida pela *The Apache*

SpamAssassin Project. A Tabela 5 apresenta os resultados com relação a Falso Positivo e Falso Negativo.

Tabela 5 - Resultado da Análise

Arquivos analisados	Falso Positivo	Falso Negativo
50 *spams* 50 **não** *spams*	7	13
Arquivos analisados	Verdadeiro Positivo	Verdadeiro Negativo
50 *spams* 50 **não** *spams*	37	43

A partir dos resultados apresentados na Tabela 5, foi aplicada a precisão de mensagens legítimas e *spams* e também o *recall* destes. A Figura 22 apresenta a precisão para mensagens legítimas e *spams*.

$$PS = \frac{37}{37+7} = \frac{37}{44} = 0{,}84 \qquad PM = \frac{43}{43+13} = \frac{43}{56} = 0{,}76$$

22(A) 22(B)

Figura 22 - Resultados Precisão de spams e mensagens legítimas

A Figura 22 apresenta o resultado da precisão sobre os e-mails legítimos e *spams*, no qual foi possível observar que o resultado da taxa de precisão sobre *spams* foi melhor que a precisão sobre e-mails válidos. Isto acontece devido a taxa de falso positivo ter sido menor que a de falso negativo, já que este cálculo retorna a taxa de que são realmente classificados corretamente dividida pela taxa dos que foram classificados corretamente mais o classificados incorretamente dentro da mesma categoria. Assim, houve um menor índice de e-mails válidos detectados como *spams* em relação aos *spams* detectados como válidos.

A Figura 23 apresenta o resultado do *recall* sobre os *spams* e e-mails válidos.

$$RS = \frac{37}{37+13} = \frac{37}{50} = 0{,}74 \qquad RM = \frac{43}{43+7} = \frac{43}{50} = 0{,}86$$

23(A) 23(B)

Figura 23 - Resultados Recall de spams e mensagens legítimas

A Figura 23 apresenta o resultado do *recall* sobre os e-mails legítimos e *spams*, no qual foi possível observar que o resultado da taxa de *recall* sobre e-mail válidos foi melhor que a precisão sobre *spams*. Isto acontece devido a taxa de falso positivo ter sido menor que a de falso negativo, já que este cálculo retorna a taxa de que são realmente classificados

corretamente dividida pelo total da mesma categoria, ou seja, o que foi classificado corretamente mais o que não foi classificado corretamente, todos da mesma categoria, no caso *spams* ou e-mails válidos. Assim, houve um menor índice de e-mails válidos detectados como *spams* em relação aos *spams* detectados como válidos.

Nota-se que os resultados da taxa de precisão e *recall* estão inversamente relacionados, ou seja, quanto melhor a precisão de uma categoria, pior será o *recall* da mesma categoria. A Tabela 6 apresenta os resultados dos cálculos.

Tabela 6 - Comparação entre Precisão e *Recall*

	Precisão	*Recall*
Mensagens Legítimas	76%	86%
Spams	84%	74%

Levando em consideração que o *recall* é uma métrica melhor para medir a detecção de *spams* do que a precisão, já que o recall calcula os classificados corretamente divididos entre todos os que são da mesma categoria. Para o recall tem-se uma taxa de erro de 14% para mensagens legítimas e de 26% de erro para mensagens *spams*, o que torna o resultado satisfatório, já que o mais importante é ter um baixo índice de Falsos Positivos e ainda quando se compara os resultados obtidos neste trabalho com filtros já existentes. Lembrando ainda que estes são resultados preliminares, em que a aplicação não passou por uma fase de aprendizado, o que aprimoraria os resultados.

Fabre (2005) fez um estudo de algumas técnicas de detecção de *spam*, no qual o autor instalou quatro diferentes ferramentas de combate a *spams* de acordo com a documentação de cada uma e as treinou. No treinamento, foram utilizadas 200 mensagens consideradas *spams* e 200 consideradas não *spams*. Os *softwares* anti-*spams* utilizados por Fabre foram: POPFile, Mozilla Mail, SpamAssassin e Bogofilter, os quais tiveram 100% de acertos em relação a classificação de e-mails válidos, porém, para a classificação de *spams*, obtiveram a seguinte taxa de falso negativos (Tabela 7):

Tabela 7- Resultados obtidos por Fabre (FABRE, 2005, p. 64)

	POPFile	Mozilla Mail	SpamAssassin	Bogofilter	**Seleção Negativa**
Taxa de falso negativo	1,9%	2,6%	3,0%	3,5%	**26%**
Taxa de falso positivo	0%	0%	0%	0%	**14%**

Assim, para efeito de comparação, a quantidade de falso negativo apresentada na Tabela 5 (13), que indica que 26% dos arquivos *spams* analisados não foram classificados corretamente, aponta que a aplicação desenvolvida teve uma diferença de 24% se comparado com o melhor resultado apresentado por Fabre. Porém, esta diferença provavelmente seria suprimida com a fase de aprendizado.

A próxima seção apresenta algumas considerações sobre os resultados, bem como alguns conceitos que não foram implementados.

4.3.1. Considerações Finais dos Resultados

De forma geral, a aplicação obteve bons resultados. Isto porque a aplicação, com o uso do Algoritmo de Seleção Negativa para detecção de *spams* aliada com a técnica de Hamming para a medida de similaridade, possibilitou a um bom resultado no que diz respeito à detecção de *spams*, tendo em vista que o algoritmo não passou pela fase de aprendizado, o que lhe daria melhores resultados.

Por isto, uma otimização através da aprendizagem de máquina em que a aplicação, através da interferência humana, possa "aprender" a identificar novos padrões, faz-se necessária. Assim, quando for fornecida para aplicação uma entrada de um e-mail válido como *spam*, este poderá ser marcado como não *spam* e a partir deste momento entraria a parte de aprendizagem.

A próxima seção apresenta as considerações finais deste trabalho bem como os trabalhos futuros relacionados.

5 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo apresentar uma solução para catalogação de *spams*. Para definição da solução, inicialmente, foram estudados os principais conceitos do Sistema Imunológico Humano e, a partir destes, o Sistema Imunológico Artificial e as teorias que validam a sua utilização: Seleção Clonal, Rede Imunológica e Seleção Negativa. O estudo de tais teorias ofereceu subsídio para escolher a teoria de Seleção Negativa como melhor modelo a ser adotado para a solução ou otimização do problema.

A princípio foi estudado sobre e-mail, agentes e protocolos de envio e recebimento de e-mails. Logo em seguida, foram abordados os conceitos relacionados a *spam*, sua origem, tipos, meios de disseminação, além das técnicas para a detecção de *spams*. Foram abordados também pesquisas acerca da identificação de *spam*, como o volume de *spams* enviados, as consequências e os perigos provenientes do *spam*. Tais conceitos tiveram importância na validação a proposta de detecção através do SIA – Sistema Imunológico Artificial. Para o entendimento do SIA, foi realizado o estudo sobre Sistema Imunológico Humano, o qual possui uma literatura abrangente e continuada, pois a espécie humana se encontra em constante evolução.

O estudo sobre o Sistema Imunológico Humano teve como base os princípios da imunidade adaptativa, que são características relevantes para serem aplicadas em resolução de problemas do mundo real, neste caso, na detecção de *spams*. As principais características são: evolução, memória, reconhecimento de padrões, detecção de anomalias. Estas características estão presentes nas teorias do Sistema Imunológico, que são: a Teoria da Seleção Clonal, Teoria da Rede Imunológica e Teoria da Seleção Negativa. Estas teorias foram adaptadas e transformadas em algoritmos, possibilitando que as mesmas sejam aplicadas na resolução ou atomização de problemas.

Tanto a Teoria da Seleção Clonal quanto a Teoria da Rede Imunológica poderiam ter sido utilizadas no desenvolvimento da aplicação, porém, a implementação destes é mais indicada para a resolução de outros problemas por possuir características relacionadas a tais problemas. Por exemplo, a implementação da Seleção Clonal é mais indicada para problemas relacionados a busca e otimização, já que este modelo tem que clonar apenas as células que são capazes de reconhecer o antígeno. Já a implementação da Rede Imunológica é mais

indicada para a clusterização de dados, pois é formada por uma rede de células que se agrupam conforme o reconhecimento e há um rearranjo a cada nova inserção.

A escolha da Teoria de Seleção Negativa fundamentou-se na distinção do próprio e não próprio, além do modelo ser o mais utilizado na área de segurança, tanto a segurança computacional (antivírus, antiSpams, sistemas de detecção de Intrusão), quanto as demais áreas, como, por exemplo, o uso de Seleção Negativa para detecção de faltas em sistemas de transmissão de energia elétrica.

Os resultados obtidos foram satisfatórios, levando em consideração que a aplicação não passou por uma fase de aprendizado. Assim, com o intuito de aprimorar os resultados, um aspecto que deve ser levado em consideração quando se fala de otimização da aplicação seria a fase de aprendizado, além de ampliar a quantidade de termos analisados. Para tanto, seria necessário uma estrutura mais complexa para a manipulação dos dados.

Uma das dificuldades encontradas no desenvolvimento da aplicação foi a fase de entendimento do algoritmo, no que diz respeito a dados próprios e não próprios, e também com relação a estrutura de utilizada para a manipulação dos dados. Superadas tais dificuldades, o desenvolvimento incluiu alguns métodos prontos que contribuíram para diminuição de classe e métodos, focando apenas nos métodos referentes ao modelo escolhido.

A utilização de Algoritmo de Seleção Negativa mostrou-se apta para a solução do problema relacionado à detecção de *spams*. O desenvolvimento completo da aplicação conduziria a melhores resultados e a uma solução adaptável para a classificação de *spams*, uma vez que a fase de aprendizado estaria sempre gerando novos detectores a cada novo *spam* fosse marcado pelo usuário. Com os resultados obtidos foi possível validar os estudos sobre o SAI através do desenvolvimento do algoritmo de Seleção Negativa para a detecção de *spams*, tal algoritmo mostrou-se apto a detectar *spams* em meio a e-mails comuns, tendo uma baixa taxa de Falsos Positivos.

A seguir são apresentados alguns trabalhos futuros, visando melhorar os resultados obtidos, além de expandir a abrangência da aplicação.

5.1. Trabalhos Futuros

Com relação a trabalhos futuros, propõem-se vários acréscimos à aplicação, os quais serão listados a seguir:

- **Aumento do número de termos analisados** – como a aplicação desenvolvida não é capaz de analisar três termos ou mais, ficando limitada apenas um ou dois termos, tem-se como proposta a análise com três ou mais termos, o que daria uma precisão maior da

categorização, já que estes termos apareceriam na sequência, logo a análise de termos sequenciais seria mais relevante do que em termos isolados. Outro ponto a ser observado é a atribuição de pesos as palavras, por exemplo, a palavra "viagra" ter um peso maior que "promoção";

- **Análise de conteúdo baseados em imagem** – a solução proposta no presente trabalho pode ser estendida para análise de imagens, em que seria possível, por exemplo, apresentar como solução a utilização do OCR (*Optical Character Recognition*) para converter o conteúdo da imagem em caracteres antes de submete-lá ao algoritmo de seleção negativa;
- **Utilização de outra medida de similaridade** – realizar um estudo comparativo da medida de Hamming com outras medidas de similaridade, como a medida do coseno ou Euclidiana, o que permitiria, por exemplo, ter ao final a definição da medida que oferece melhor desempenho para o algoritmo;
- **Análise semântica do conteúdo** - Com o advento da Web semântica seria interessante realizar uma análise do sentido do texto. Assim, para análise semântica das palavras poderia ser utilizado o *Jena* (*Framework Web* Semântico em Java);
- **Implementação do aprendizado** – um ponto importante para continuidade do trabalho seria a definição de um ambiente de aprendizagem do algoritmo de seleção negativa, no qual seria detectado um dado que não havia sido detectado antes (Falso Negativo) e/ou deixaria de detectar o que havia sido detectado (Falso Positivo). Este aprendizado poderia ocorrer a partir de inferência humana, no qual seria marcado se o dado é ou não *spam*. A partir dessa inferência os detectores que definiram um e-mail válido como *spam*s seriam excluídos e para os *spams* que foram classificados como válidos seriam gerados novos detectores. Para esta fase de aprendizado sugere-se a utilização de APIS ou *frameworks* para implementação da solução que geraria novos detectores. Uma possibilidade seria a utilização de uma API que implemente o Algoritmo Genético, que, a partir dos cruzamentos e mutações definidos por ele, torne possível a geração de novos detectores;
- **Teste em ambiente real** – Realizar testes em um ambiente real, no qual a aplicação seria inserida como um filtro dinâmico para a detecção dos e-mails que passaram pelos filtros estáticos.

A abordagem destas possibilidades de trabalhos futuros tem por objetivo a otimização dos resultados já obtidos e expandir a abrangência dos mesmos.

6 REFERÊNCIAS BIBLIOGRÁFICAS

Antispam.br. Disponível em : <http://www.antispam.br>. Acesso em 15 de setembro de 2011.

ALMEIDA, Tiago Agostinho; YAMAKAMI, Akebo; TAKAHASHI, Márcia Tomie. Sistema imunológico artificial para resolver o problema da árvore geradora mínima com parâmetros fuzzy. **Pesquisa Operacional**, Rio de Janeiro, v. 27, n. 1. 2007 . Disponível em <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382007000100008&lng=en&nrm=iso>. Acessado em 03 Abril de 2011. DOI: 10.1590/S0101-74382007000100008.

ALMEIDA, Thiago Tassar de. **Estudo de Caso**: Implementação De Um Serviço De E-Mail Para O Departamento De Computação. Ouro Preto, 2010. Monografia (Ciência da Computação). Universidade Federal de Ouro Preto. Disponível em: <http://www.decom.ufop.br/menotti/monoII102/files/BCC391-102-vf-04.1.4293-ThiagoTassarDeAlmeida.pdf>. Acesso em: 05 de novembro de 2011.

AMARAL, Jorge Luís Machado do. **Sistemas Imunológicos Artificiais Aplicados à Detecção de Falhas**. Rio de Janeiro, 2006. Tese (Doutorado). PUC– Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica. Disponível em: < http://www2.dbd.puc-rio.br/pergamum/tesesabertas/> Acesso em: 23 de maio de 2011.

ASSIS, João Marinho de Castro**. Detecção de E-mails Spam Utilizando Redes Neurais Artificiais.** Itajubá-MG, 2006. Dissertação (Mestrado). UNIFEI – Universidade Federal de Itajubá.

BALACHANDRAN, Sankalp. **Multi-shaped Detector Generation Using Real Valued Representtion for Aanomaly Detection**. Memphis, 2005. Tese (Doutorado). University of Memphis. Disponível em: <http://ais.cs.memphis.edu/files/papers/SankalpThesisFinal.pdf>. Acesso em: 02 de junho de 2011.

BERBERT, Priscila Cristina; **Sistema Imunológico Artificial para Otimização Multiobjetivo.** Campinas-SP, 2008. Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Departamento de Telemática. Disponível em <http://cutter.unicamp.br/document/?code=vtls000440407 > acesso em 16 de fevereiro de 2011.

CASTRO, Leandro Nunes de. **Engenharia Imunológica:** Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais. Campinas-SP, 2001. Tese (doutorado). Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e Computação. Disponível em: <http://www.dca.fee.unicamp.br/~vonzuben/research/lnunes_dout/index.html > Acesso em: 18 de março de 2011.

CASTRO, Leandro Nunes de; VON ZUBEN, Fernando José. **An Evolutionary Immune Network for Data Clustering.** In Proceedings of the IEEE SBRN (Brazilian Symposium on Artificial Neural Networks), pp. 84-89, Rio de Janeiro, 2000.

CASTRO, Leandro. Nunes de; VON ZUBEN, Fernando. José**.** Learning and Optimization Using the Clonal Selection Principle. ***IEEE Transaction on Evolutionary Computation***, Special Issue sobre Sistemas Imunológicos Artificiais. 2001.

CARVALHO, Eduardo Enrique Ostos. **Raciocínio Baseado em Casos:** Uma Abordagem Utilizando o Sistema Imune Artificial. Belo Horizonte-MG, 2009. Dissertação (Mestrado em Engenharia Elétrica). Universidade Federal de Minas Gerais. Disponível em: < http://cpdee.ufmg.br/defesas/54M.PDF>. Acesso em 21 de outubro de 2011.

COPPIN, Bem. **Inteligência Artificial**. Rio de Janeiro: LTC, 2010.

DRAGO, Idilio. **Estudo comparativo de algoritmos de classificação em bases de dados com atributos temporais.** Vitória-ES, 2007. Dissertação (Mestrado) – Universidade Federal do Espírito Santo, Centro Tecnológico. Disponível em: < http://ewi1438.ewi.utwente.nl/home/sites/default/files/dissertacao.pdf >. Acesso em 5 de maio de 2012.

FABRE, Recímero César. **Métodos Avançados para Controle de Spam**. Campinas, 2005. Dissertação (Mestrado profissional) - Universidade Estadual de Campinas, Instituto de Computação. Disponível em: <http://www.las.ic.unicamp.br/paulo/teses/20050215-MP-Recimero.Cesar.Fabre-Metodos.avancados.para.controle.de.Spam.pdf>. Acesso em: 13 de junho de 2012.

FERRON, Myriam; RANCANO, Jordi. **Grande Atlas Do Corpo Humano:** Anatomia/Histologia/Patologias**.** Barueri, SP: Manole, 2007.

PROCEEDINGS OF THE CONGRESS ON EVOLUTIONARY COMPUTATION, 2002, Hawaii. **Combining Negative Selection and Classification Techniques for Anomaly Detection.** Hawaii, Maio 2002. p. 705-710.

GUZELLA, Thiago dos Santos; SANTOS, Tomaz A. Mota; UCHOA, Joaquim Q.; Caminhas, Walmir. M.. Identification of *SPAM* messages using an approach inspired on the immune system. **Biosystems**, v. 92, p. 215-225, 2008.

5TH INTERNATIONAL CONFERENCE ON ARTIFICIAL IMMUNE SYSTEMS. 2006, Portugal. **Modelagem de um Sistema Imune Artificial para a identificação de *SPAM***. Portugal, 2006. p. 09-22.

KUROSE, James F.; ROSS, Keith W. **Redes de Computadores e a Internet**: uma abordagem top-down. São Paulo: Pearson Education. 2010.

LINDEN, Ricardo. **Algoritmos Genéticos:** Uma importante ferramenta da inteligência computacional. 2. ed. Rio de Janeiro: Brasport, 2008.

RFC 822. Crocker, David H. **Standard for the Format of ARPA Internet Text Messages**. Dept. of Electrical Engineering. University of Delawar, Agosto de 1982.

TAVEIRA, Danilo Michalczuk. **Mecanismo Anti-*Spam* Baseado em Autenticação e Reputação.** 2008. Dissertação (mestrado) - Universidade Federal do Rio de Janeiro, Rio de Janeiro. Disponível em: < http://www.gta.ufrj.br/ftp/gta/TechReports/Danilo08.pdf>. Acesso em: 22 de setembro de 2011.

TAVEIRA, Danilo Michalczuk; MORAES, Igor Monteiro; RUBINSTEIN, Marcelo Gonçalves; DUARTE, Otto Carlos Muniz Bandeira. **Técnicas de Defesa Contra *Spam.*** in Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, 2006, Santos. **Minicursos.** Rio de Janeiro: Universidade Federal do Rio de Janeiro. p. 202-250.

MICHELAN, Roberto. **Evolução de redes imunológicas para coordenação automática de comportamentos elementares em navegação autônoma de robôs**. Campinas, SP. 2003. Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação. Disponível em: < http://cutter.unicamp.br/document/?code=vtls000385823&fd=y> Acesso em: 29 de maio de 2011.

ODA, Terri; WHITE, Tony. Spam Detection using an Artificial Immune System. **Revista Crossroads**, Nov. 2004. Disponível em < http://terri.zone12.com/doc/academic/crossroads/>. Acessado em: 25 de outubro de 2011.

PARHAM, Peter. **O Sistema Imune**. Porto Alegre: Artmed, 2001.

RUSSELL, Stuart; NORVIG, Peter. **Inteligência Artificial**. Rio de Janeiro: Campus, 2004,

SILVA, Guilherme Costa**. Detecção de Intrusão em Redes de Computadores**: Algoritmo Imunoinspirado Baseado na Teoria do Perigo e Células Dendríticas. Belo Horizonte, 2009. Dissertação (mestrado) – Universidade Federal de Minas Gerais.

SCRIMGER, Rob; LASALLE, Paul; PARIHAR, Mridula; GUPTA, Meeta. **TCP/IP - A Bíblia.** Rio de Janeiro: Campus, 2002.

Spam Statistics. **M86 Security Labs**. 18 Set. 2011. Disponível em: <http://www.m86security.com/labs/spam_statistics.asp>. Acesso em 21 de setembro de 2011.

Symantec Intelligence Report. **Symantec**. Jul, 2011. Disponível em: < http://www.symantec.com/content/en/us/enterprise/other_resources/b-intelligence_report_07-2011.en-us.pdf >. Acesso em: 18 de setembro de 2011.

SZENDRODI, Rafael Jorge Csura; BANDEIRA, Otto Carlos Muniz. **O *SPAM*:** suas origens, sua evolução e as técnicas para evitá-lo. 2005. Disponível em: <http://www.gta.ufrj.br/grad/05_1/spam/szendro/index.html>. Acesso em: 21 de setembro de 2011.

VON ZUBEN, Fernando José. **O mundo natural e o mundo artificial.** Universidade Estadual de Campinas, Departamento de Computação Engenharia e Automação, Faculdade de Engenharia Elétrica e de Computação. 2011. Disponível em: < ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia707_1s11/notas_de_aula/topico1_IA707_1s11.pdf>. Acesso em: 20 de maio de 2011.