UNIVERSIDADE FEDERAL DE ITAJUBÁ

PROGRAMA DE PÓS GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Implementação e Validação de Novos Módulos em um Sistema Anti-Spam

Anthony Miranda Vieira

Itajubá, Novembro de 2014

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Anthony Miranda Vieira

Implementação e Validação de Novos Módulos em um Sistema Anti-Spam

Dissertação submetida ao Programa de Pós-Graduação em Ciência e Tecnologia da Computação como parte dos requisitos para obtenção do Título de Mestre em Ciência e Tecnologia da Computação

Área de Concentração: Sistemas de Computação

Orientador: Prof. Dr. Otávio Augusto Salgado Carpinteiro
Co-Orientador: Prof. Dr. Edmilson Marmo Moreira

Novembro de 2014
Itajubá - MG

Agradecimentos

Agradeço a Deus por direcionar a minha vida nos melhores caminhos, por permitir que aprenda com as dificuldades e valorize as alegrias. Meus agradecimentos se estendem à Universidade Federal de Itajubá, ao meu orientador, co-orientador e demais membros do corpo docente do programa de mestrado em Ciência e Tecnologia da Computação, não só pelo apoio e incentivo na realização do mesmo, mas pela disposição e dedicação ao trabalho nobre de lecionar. Agradeço também o aluno de graduação Isaac Caldas Ferreira pela grande ajuda prestada durante todas as etapas deste trabalho. Por fim, aos amigos e familiares pelo carinho e ajuda nos momentos delicados, a paciência e aceitação nos momentos que não estive presente e pelo incentivo a conclusão desta dissertação.

A persistência é o caminho do êxito.

Charles Chaplin

Resumo

O correio eletrônico é uma das principais formas de comunicação. O maior problema encontrado atualmente em sua utilização é o crescente número de mensagens indesejadas (*spams*) recebidas diariamente pelos usuários. O grande volume de *spams* causa prejuízos, tais como, desperdícios de tempo, de espaço de armazenamento, da largura da banda de rede, bem como comprometimento no recebimento de mensagens, atrasos, disseminação de vírus, *spybots*, dentre outros. É importante que sejam desenvolvidas ferramentas e técnicas de combate a esta prática com intuito de mitigar estes problemas.

O desafio do problema em questão reside no fato de que os sistemas (ou filtros) anti-*spam* evoluem através de técnicas de detecção e bloqueio eficazes e, em contrapartida, os *spammers* criam e desenvolvem novas técnicas de ofuscamento para burlar tais sistemas.

Esta dissertação aborda a implementação e validação de novos módulos em um sistema anti-*spam* (SAS) que emprega técnicas de análise de conteúdo e redes neurais. O sistema é composto por um novo pré-filtro, que faz uso de métodos, desenvolvidos neste trabalho, para combate ao ofuscamento de conteúdo, por um módulo de seleção de características, que analisa o conteúdo da mensagem buscando palavras relevantes para redução da complexidade da classificação e, por fim, por um novo modelo neural MLP (*Multilayer Perceptron*), implementado em Java e treinado com *backpropagation*, para classificar os *e-mails* em duas classes — *ham* e *spam*.

Os testes foram realizados no ambiente real da Universidade Federal de Itajubá e comparados com o desempenho de um filtro anti-*spam* de uso comercial (Barracuda) utilizado na universidade. Foram empregadas três técnicas de seleção de características, com diferentes combinações de características. Os resultados obtidos sÃ£o promissores.

PALAVRAS-CHAVE: *Spam*, *E-mail*, *Ham*, Redes Neurais Artificiais, MLP, Filtro Anti-*Spam*, *Backpropagation.*

Abstract

Currently, e-mail is one of the most important ways of communication. The major problem in its usage is the continuous raising of undesirable messages received daily by users. The amount spam's volume leads to several losses, such as, time waste, extra storage spaces, bandwidth loss, delay, non-transmission of valid messages, virus, spybot spread, among other things. It is important to develop tools and techniques in order to mitigate these problems.

The key challenge lays on the fact that anti-spam filters evolve through new and effective detection and block methods and spammers, on the other hand, keep creating obfuscation techniques to avoid them.

This dissertation is about an implementation and validation of new modules of an anti-spam system (SAS) which applies content analysis and artificial neural networks technique. This system was made by a pre filter composed by methods developed in this dissertation to avoid content obfuscation technics, allied to a characteristic selection module that analyses the message's content to search for keywords to reduce the classification complexity and a new neural model MLP, codified in Java and trained in backpropagation which classifies the e-mail in two classes — ham or spam.

The tests were realized in the real environment of the Federal University of Itajubá and the results were compared to a commercial anti-spam filter (Barracuda) used by this University. There were applied three different techniques to characteristic selection and different combinations of the input vectors. Hence, the attained results are promising.

KEYWORDS: Spam, E-mail, Ham, Artificial Neural Networks, MLP, Anti-Spam Filter, Backpropagation.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ANN	*Artificial Neural Network*
CBT	*Corpus Based Thesaurus*
CTSS	*Compatible Time-Sharing System*
DF	*Distribuição de Frequência*
DoS	*Deny of Service*
E-mail	*Electronic Mail*
GA	*Genetic Algorithm*
GPESC	*Grupo de Pesquisas em Engenharia de Sistemas e de Computação*
HSS	*Hybrid Semantic Similarity*
HTML	*HyperText Markup Language*
IP	*Internet Protocol*
ISO	*International Standardization Organization*
LMS	*Learning Management Systems*
LQV	*Learning Vector Quantization*
LSA	*Latent Semantic Analysis*
LSFS	*Latent Semantic Feature Space*
MCP	*McCulloch e Pitts*
MI	*Mutual Information*
MIT	*Massachusetts Institute of Technology*
MLP	*Multilayer Perceptron*
NIC	*Network Information Center*
RFC	*Request for Comments*
SAS	*Sistema Anti-Spam*
SVM	*Support Vector Machine*
TCP	*Protocolo de Controle de Transmissão*
TCR	*Total Cost Ratio*
UTF	*Unicode Transformation Format*
URL	*Uniform Resource*
XML	*eXtensible Markup Language*

1 Introdução

1.1 Considerações Iniciais

Nos dias atuais, tem-se como um dos principais meios de comunicação o *e-mail*, palavra de origem inglesa abreviada de *electronic mail*. Trata-se de um serviço, disponível na rede mundial de computadores (*Internet*), que tem por finalidade prover comunicação entre pessoas desta rede (VAN VLECK, 2001). Esta forma de comunicação tem sido um dos principais avanços sócio tecnológicos dos últimos 40 anos (PARTRIDGE, 2008).

A evidência do impacto causado, na sociedade atual, pela *Internet* e por suas formas de comunicação é mostrada por meio de um levantamento realizado em julho de 2012. Este indicou que o tempo gasto por um profissional utilizando um serviço de correio eletrônico, entre leitura e envio de mensagens, é cerca de 28% de seu dia. (MCKINSEY, 2012).

Este capítulo visa abordar a importância e características dos *e-mails*, o surgimento de mensagens não solicitadas, os tipos e problemas gerados pelas mesmas, as tecnologias criadas para evitar estas mensagens e, por fim, a proposta deste trabalho.

1.2 Importância e característica dos E-mails

Estudos de 2012 indicaram que por dia circulam na *Internet* cerca de 144 bilhões de *e-mails* (ROYAL PINGDOM, 2013). O envio efetivo e regular de *e-mails*, além de um meio de comunicação e transferência de arquivos, é importante para qualquer abordagem empregada atualmente na *Internet*, como, por exemplo, para atrair visitantes para *websites*, conquistar a fidelidade dos usuários, divulgação de propagandas de produtos, informa-

tivos, dentre outros (MCDONALD, 2009). As principais características que alavancam todo este montante diário de *e-mails* são:

- *E-mails* são rápidos: No mundo globalizado, as informações são dependentes do tempo, com atualizações e descobertas constantes. Precisam ser entregues em questão de minutos e não dias ou semanas (LI *et al.*, 2006).
- *E-mails* são simples: O endereço eletrônico de um usuário é mais facilmente compartilhado que seu endereço físico. Possui vantagens em relação à segurança da informação, à facilidade de resposta e à mobilidade, ou seja, acesso à mensagem independente de uma estrutura fixa, podendo ser acessada de qualquer dispositivo compatível com acesso à *Internet* (LI *et al.*, 2006).
- *E-mails* são versáteis: Permitem o envio de qualquer tipo de arquivo via rede, como, por exemplo, imagens, textos, vídeos, *links*, dentre outros (LI *et al.*, 2006).
- *E-mails* geram respostas imediatas: Mensagens que possuam *links* possibilitam acesso direto a *sites*, registros em comunidades e compras. Portanto, geram resultados que podem ser mensurados instantaneamente (MCDONALD, 2009).
- *E-mails* são direcionados: Com as atuais caixas de correio e sistemas de envio, as listas de usuários de *e-mail* podem ser segmentadas em grupos e, desta forma, permitem que as mensagens atinjam determinados tipos de destinatários de acordo com a necessidade do emissor (LI *et al.*, 2006).
- *E-mails* são pró ativos: Mensagens eletrônicas permitem abordagem direta ao usuário, informando-o a respeito de produtos ou notícias de sua preferência, sem a necessidade do mesmo buscar tais informações (MCDONALD, 2009).
- *E-mails* atingem mais destinatários: As caixas de correio permitem que os destinatários reencaminhem os *e-mails* recebidos a outros destinatários, o que aumenta o poder de alcance da mensagem original (MCDONALD, 2009).

No entanto, as vantagens obtidas pelos *e-mails* serviram também para o surgimento do envio de mensagens não solicitadas no meio eletrônico. As próximas seções trazem informações a respeito deste problema.

1.3 Mensagens não Solicitadas

Como pode ser visto na seção 1.2, os *e-mails* apresentam grandes vantagens e avanços na comunicação entre as pessoas. Estas facilidades, porém, incentivam também a proliferação de mensagens não desejadas na rede mundial de computadores. Estas mensagens são denominadas *e-mails spam* ou simplesmente *spams* (MUELLER, 2013).

Não existe uma definição formal para o termo *spam*, devido ao fato de sua abrangência e de suas formas de manifestação estarem em constante adaptação. Contudo, uma das descrições mais apropriadas é a de que um *spam* é *o ato de enviar ou publicar, na Internet, inúmeras cópias da mesma mensagem (ou com conteúdo muito similar) com o intuito de atingir os destinatários que não as querem receber* (MUELLER, 2013).

A origem do primeiro *spam* data de três de maio de 1978. Foi enviado por Gary Thuerk para 393 pessoas via *Arpanet* (NPR, 2008). Este é considerado o primeiro *e-mail* em massa não solicitado, enviado com o intuito de promoção de vendas sem o interesse prévio dos destinatários.

A origem da alcunha *SPAM*, como significado de mensagem não solicitada, data de abril de 1994, quando dois advogados, Canter e Siegel, da cidade de Phoenix nos Estados Unidos, publicaram uma propaganda de seus serviços. Para tal, eles contrataram um programador para a criação de um *script* que enviava a mensagem para todos os grupos da *Usenet*, até então o maior sistema de conferência *online* do mundo (CAPANEMA, 2009).

Quando os usuários receberam a propaganda dos advogados, associaram o fato a um episódio de programa humorístico inglês chamado *Monty Phyton Flying Circus*. No programa, um restaurante serve todos os seus pratos de comida com muito SPAM[1] e a garçonete repete a palavra diversas vezes para demonstrar o quanto de SPAM vem no prato. Quando ela faz isto, um grupo de vikings sentados no canto de uma mesa começa a cantar *"spam, spam, spam, spam, spam, lovely spam! Wonderfull spam!"* (CAPANEMA, 2009). O indivíduo que espalha mensagens *spam* pela *Internet* ficou popularmente conhecido como *spammer*.

[1]SPAM é um alimento feito de carne pré-cozida e enlatada pela empresa Hormel Foods Corporation

Grande parte dos *spams* apresenta características recorrentes, como as descritas a seguir:

- São procedentes de emissores desconhecidos do usuário, enviados por endereços de *e-mail* fictícios ou criados exclusivamente para o envio das mensagens (TEIXEIRA, 2004).
- O anúncio em si é pífio, fraudulento e/ou ilegítimo. Possui textos chamativos, ameaças e *links* para acesso a *sites* externos. As mensagens não identificam claramente o grupo ou as pessoas que as enviaram (TEIXEIRA, 2004).
- Possuem um endereço eletrônico para exclusão, ou seja, o receptor deve mandar um *e-mail* para este endereço caso deseje não receber futuras mensagens desta fonte. Entretanto, esta opção é falsa, porque mesmo depois de enviar o *e-mail* para o endereço, o usuário continua recebendo *spams* desta fonte (TEIXEIRA, 2004).

A Figura 1 ilustra um exemplo típico de *spam* onde algumas das características descritas anteriormente podem ser identificadas.

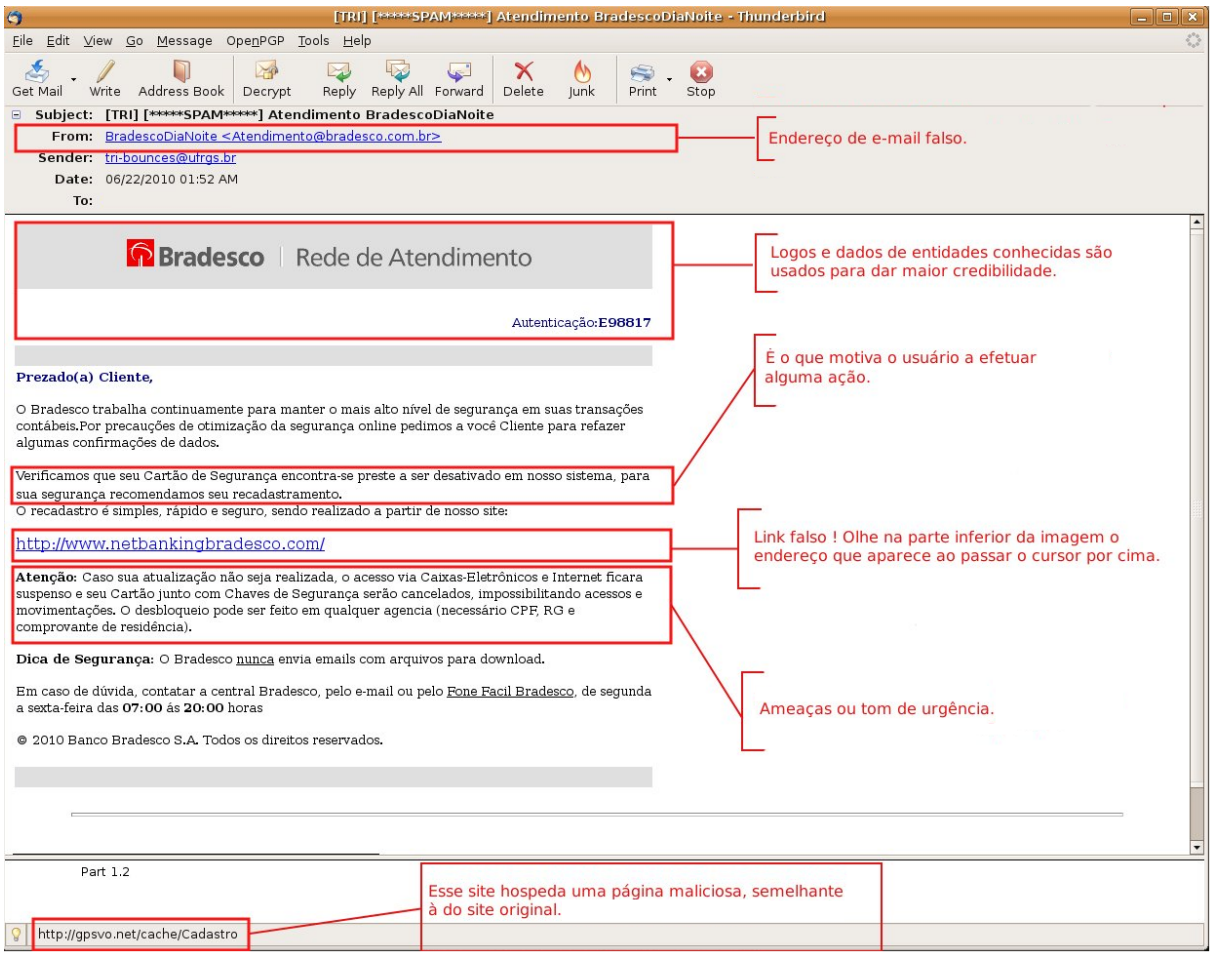

Figura 1: *E-mail spam com características típicas*

1.4 Problemas causados por Spams

Estima-se atualmente que 68,8% de todo o tráfego de *e-mails* na rede seja de *spams* (ROYAL PINGDOM, 2013). Baseado nesta estatística, conclui-se que mais de dois terços das mensagens enviadas via *Internet* não são solicitadas. O impacto deste volume na rede gera diversos problemas para todos os usuários envolvidos direta ou indiretamente com as mensagens (CAPANEMA, 2009).

A seguir, estão listados os principais problemas relacionados às mensagens não solicitadas:

- "Corrida grátis": Este termo ironiza o custo implícito de uma passagem gratuita. Quando *e-mails spam* são enviados, os destinatários pagam mais por eles do que os emissores. Por exemplo, a AOL reportou em 2009 que estava recebendo cerca de 1,8 milhões de *spams* de promoções *online* por dia (BACHMANN *et al.*, 1996). Pressupondo que leve cerca de 10 segundos para que um usuário identifique a mensagem e a descarte, são gastas cinco mil horas por dia de tempo conectado descartando-se *e-mails*. Por outro lado, o dia de trabalho de um *spammers* custa em torno de 100 dólares (BACHMANN *et al.*, 1996). Logo, todos os custos implícitos de tempo, de uso do servidor de *e-mails*, de conexão à *Internet*, dentre outros, são arcados pelo destinatário.

- "Oceanos de *spam*": A quantidade de mensagens não solicitadas é facilmente escalável através dos recursos presentes no próprio serviço de mensagem eletrônica. Desta forma, um *e-mail* encaminhado para uma lista de usuários e, posteriormente, encaminhado por estes usuários, consegue atingir progressões aritméticas de propagação (BACHMANN *et al.*, 1996). Por isto, usuários da AOL em 2012 relataram que estão atingindo níveis elevados de mensagens (cerca de 100 *spams* diários) recebidos dos próprios usuários da AOL, o que tem prejudicado a leitura dos *e-mails* relevantes (BACHMANN *et al.*, 1996).

- Roubo de recursos: Um número crescente de *spammers* envia a maioria de suas mensagens através de sistemas intermediários "inocentes"(servidores comuns de *e-mail*), ao invés do envio por servidores específicos de *spam*, com o intuito de burlar os mecanismos de filtragem dos sistemas anti-*spam*[2]. Este procedimento sobrecarrega o tráfego e o armazenamento dos sistemas intermediários, fazendo com que os responsáveis por estas redes percam seu tempo lidando com os *spams*. Isto também prejudica a conexão do usuário final com a rede (CAPANEMA, 2009). Uma técnica muito utilizada, por exemplo, é chamada de "bater e correr". Os *spammers* recebem acesso *trial*[3] dos servidores, aproveitam para enviar grandes quantidades de *e-mail* e abandonam a conta posteriormente. Este processo faz com que os responsáveis pelos servidores gastem mais recursos mantendo equipes para mantê-los operantes e eficientes (CAPANEMA, 2009).

- Sem conteúdo: As mensagens de *spam*, em sua grande maioria, tratam de propaganda de produtos sem valor, insignificantes, ou parcial ou totalmente fraudulentos.

[2]Filtros de *spam* são abordados no subcapítulo 1.7.

[3]É um serviço fornecido pelo servidor que permite ao usuário acesso gratuito por algum tempo limitado para que o mesmo possa testar o serviço.

Como o custo de espalhar *spams* é bem inferior ao custo de uma propaganda normal, certas empresas optam por eles. As mensagens irrelevantes podem também visar capturar dados e informações dos usuários para aumentar o ganho ilegal dos *spammers* mal intencionados (CAPANEMA, 2009).

- Ilegalidade: Certos tipos de *spam* são ilegais em determinados países, especialmente os que envolvem pornografia e sistemas fraudulentos (GOETSCHI, 2004).

- Falta de ética: Receber *spams* não é uma atividade opcional. Os *softwares* que os enviam geralmente possuem uma lista de nomes aleatórios de pessoas colhidos de listas de *e-mail* e são desenvolvidos para não serem detectados (GOETSCHI, 2004). Para que a mensagem seja facilmente aceita, os *spammers* utilizam, como endereços de origem, endereços falsos de *e-mail* similares a endereços conhecidos de lojas virtuais famosas (GOETSCHI, 2004).

1.5 Combate aos Spams

Conforme descrito nas subseções anteriores, os *spams* tornaram-se um problema para toda a comunidade que utiliza a *Internet*. O número de mensagens não solicitadas em massa cresce com o passar dos anos (ROYAL PINGDOM, 2013). Diversas técnicas já foram e ainda são empregadas na tentativa de solucionar estes problemas, mas nenhuma de forma totalmente eficaz (TEIXEIRA, 2004).

Existem duas principais abordagens para combater *spams*. A primeira opera no lado do emissor da mensagem. São técnicas projetadas para prevenir que usuários mal intencionados enviem os *spams*. Esta abordagem é a ideal, pois evita o tráfego das mensagens *spam* na rede e, com isto, previne também os problemas provenientes das fontes de *spam*. Entretanto, devido à distribuição massiva das fontes de *spam* e a existência de "*spam-bots networks*"[4], esta é a abordagem mais difícil de ser implementada (PARK *et al.*, 2005).

Portanto, o combate ao *spam* normalmente emprega a segunda abordagem, a que atua no lado do receptor da mensagem. Sua atuação pode ocorrer no servidor de *e-mail*, no provedor de serviços de *Internet*, ou no cliente de *e-mail* (TAKASHITA *et al.*, 2008). Os

[4]Computadores comprometidos (em qualquer lugar do mundo) que podem enviar milhões de *spams* diariamente.

sistemas anti-*spam* implementam esta abordagem. A grande dificuldade presente nesta abordagem é devida ao surgimento dos falsos positivos e negativos (abordados nas seções 1.5.2) (TAKASHITA *et al.*, 2008).

Os filtros anti-*spam* visam identificar previamente o *e-mail* recebido pelo usuário na tentativa de classificá-lo como um *e-mail* válido (também conhecido como *ham*) ou como inválido, o usual *spam*. Para esta classificação, existem diversas técnicas que variam em complexidade e em desempenho (TAKASHITA *et al.*, 2008).

Tais técnicas são classificadas em estáticas ou dinâmicas, conforme abordam a classificação dos *e-mails*. Listas brancas, listas negras, confirmação de envio, algoritmos *Naïve Bayesian*, árvores de decisão, redes neurais são exemplos destas técnicas (TAKASHITA *et al.*, 2008).

1.5.1 Técnicas Estáticas e Dinâmicas

As técnicas estáticas fazem uso de reconhecimento de padrões que compartilhem dos mesmos atributos e possuam as mesmas origens. Para seu emprego, faz-se necessária a intervenção do usuário (PARK *et al.*, 2005).

Listas de endereços eletrônicos criadas pelos usuários bem como características por eles selecionadas no corpo do *e-mail*, por exemplo, provêm as informações relevantes para que as técnicas estáticas possam atuar corretamente. Um exemplo de técnicas estáticas são as listas negras. Tais listas são preenchidas com endereços eletrônicos que comumente enviam *spams* e, de forma análoga, listas brancas possuem endereços de usuários confiáveis que enviam *hams* (CRIVISQUI, 1998).

Uma vez criadas as listas, o servidor de *e-mails* verifica se o endereço eletrônico do emissor consta em uma das listas e se o conteúdo da mensagem possui as características comumente selecionadas pelos usuários. Desta forma, o *e-mail* é movido automaticamente para a pasta de lixo eletrônico (*spam*) (TAKASHITA *et al.*, 2008).

A grande vantagem deste método é que, uma vez criadas as listas e cadastradas as características, nenhum esforço adicional é exigido do usuário. Em contrapartida,

a grande desvantagem provém do fato de que os *spammers* estão sempre variando as características dos *e-mails spam*. Assim, o usuário deve estar constantemente cadastrando novas características e alimentando as listas para que o sistema anti-*spam* permaneça eficaz (TAKASHITA *et al.*, 2008).

As técnicas dinâmicas levam em consideração o conteúdo dos *e-mails*, a fim de classificá-los como *ham* ou *spam*. Os servidores de *e-mail* analisam as mensagens comparando-as com uma base de dados, dinamicamente atualizada, para, então, classificá-las (PARK *et al.*, 2010).

Métodos probabilísticos, estruturas de decisão e redes neurais são exemplos de métodos dinâmicos (CRIVISQUI, 1998). Todos, quer através de aprendizado, quer de regras combinatórias, quer de armazenamento de experiências, dentre outros, utilizam uma base prévia com capacidade evolutiva, para constante aprimoramento. Este aprimoramento baseia-se nos próprios resultados obtidos em classificações anteriores, corretas ou incorretas, de *e-mails* (TAKASHITA *et al.*, 2008).

A grande vantagem deste método é o constante aprimoramento na classificação de *e-mails*. A desvantagem reside na maior incidência de casos de falsos positivos (PARK *et al.*, 2010), descritos a seguir.

1.5.2 Falsos Positivos e Negativos

À primeira vista, as técnicas dinâmicas de classificação de *e-mails* implementadas pelos servidores de mensagens eletrônicas aparentam ser a solução ideal pelo fato de se aprimorarem constantemente. Entretanto, não são técnicas plenamente confiáveis, pois há casos onde o classificador detecta padrões de *spam* em conteúdo de mensagens *ham* e classifica-as erroneamente como *spam*. Estes casos são chamados de falsos positivos (TAKASHITA *et al.*, 2008). Em suma, falsos positivos são *e-mails ham* classificados incorretamente como *e-mails spam* (TAKASHITA *et al.*, 2008). Falsos positivos são um problema grave, pois os usuários passam a verificar frequentemente seus *spams* à procura de *e-mails* legítimos, tornando ineficaz a utilização das técnicas dinâmicas. Os falsos positivos podem tornar-se um problema ainda mais grave quando os *spams* são removidos automaticamente pelo sistema anti-*spam*, sem a notificação ao usuário (PARK *et al.*,

2005).

O oposto também ocorre e deve ser evitado. São casos onde *e-mails spam* apresentam fortes características de mensagens legítimas e, portanto, são classificados como *hams* (PARK *et al.*, 2005). Estes erros de classificação são conhecidos como falsos negativos. São menos prejudiciais comparados aos casos de falsos positivos, pois os próprios usuários removem os *e-mails* identificados como *spam* presentes em suas caixas de entrada.

1.6 Proposta do Trabalho

Este trabalho visa desenvolver e validar novos módulos para um Sistema Anti-*Spam* (SAS) que vem sendo desenvolvido no GPESC (Grupo de Pesquisas em Engenharia de Sistemas e de Computação). As etapas a seguir descrevem os objetivos traçados.

Primeiro, desenvolver e validar um novo módulo para abordar uma gama complexa de novas técnicas de ofuscamento de *e-mail*. Estas técnicas foram desenvolvidas pelos *spammers* com a finalidade de burlar os filtros da atualidade. Após a análise do conteúdo do *e-mail*, as técnicas são identificadas e marcadas (seção 4.3). Segundo, desenvolver um novo modelo neural MLP (*Multilayer Perceptron*) na linguagem Java, uma vez que o modelo neural anterior (MLP) encontrava-se implementado em MatLab. Terceiro, integrar todos os módulos do SAS em um sistema unificado. Quarto, definir um arquivo de configuração, no formato XML. Assim, as funcionalidades do SAS podem ser desabilitadas ou habilitadas de acordo com a necessidade do usuário. Quinto, comparar o desempenho do SAS com o do sistema comercial Barracuda.

1.7 Considerações Finais

Este Capítulo abordou a importância das mensagens eletrônicas no contexto atual, bem como o surgimento das mensagens não solicitadas, que vieram a ser intituladas de *spams*. Abordou, igualmente, os problemas causados por *spams* e os mecanismos para seu combate.

O Capítulo 2 apresenta uma revisão da teoria que fundamenta o trabalho desenvolvido. No Capítulo 3, são discutidas as propostas existentes para filtragem de *e-mails*. O Capítulo 4 traz a técnica empregada nos estudos e os resultados obtidos. Por fim, o Capítulo 5 conclui a dissertação e aponta novas diretrizes para pesquisas futuras.

2 Revisão Teórica

2.1 Considerações Iniciais

Neste capítulo, aborda-se a teoria que fundamenta este trabalho. São detalhados os elementos constituintes envolvidos na composição dos *e-mails* e no tráfego de mensagens eletrônicas na *Internet*, as técnicas de ofuscamento presentes no envio e no corpo das mensagens, os métodos de seleção de características em textos e, finalmente, os modelos neurais artificiais.

2.2 E-mails e Servidores

A seguir, uma breve revisão sobre a composição e funcionamento dos *e-mails* e servidores.

2.2.1 E-mail

E-mails são as mensagens enviadas eletronicamente, através de *softwares* servidores específicos, e acessadas através de *softwares* clientes de *e-mail*. Como toda mensagem transmitida, eles possuem um endereço de origem (emissor da mensagem) e um endereço de destino (receptor). São compostos basicamente pelo corpo, cabeçalho e envelope.

O corpo do *e-mail* contém a mensagem e anexos. O texto da mensagem pode estar presente de forma simples, conhecido como *plain text*, que não precisa ser interpretado pelo cliente de *e-mail*, ou pode estar no formato HTML, com *tags* de marcação que precisam

ser processadas pelo cliente para então serem exibidas ao usuário.

O cabeçalho contém informações de controle da mensagem como, por exemplo, assunto, endereço de retorno (pode ser copiado do envelope), demais endereços para onde a mesma foi encaminhada, data, hora, endereço do emissor, dentre outras.

O envelope contém os endereços do emissor e do receptor da mensagem. Estes são utilizados pelo servidor de *e-mails* para identificar a origem e o destino da mesma. Possui também o endereço para retorno de uma mensagem de erro, caso o *e-mail* não alcance seu destino.

2.2.2 Servidor de E-mail

Servidores de *e-mail* são aplicações responsáveis por receber, enviar e armazenar mensagens eletrônicas de usuários presentes na mesma rede local. Computadores dedicados à execução destas aplicações são também conhecidos como servidores de *e-mails*. Os servidores monitoram portas TCP (Protocolo de Controle de Transmissão) específicas, aguardando conexões de usuários.

De forma simplificada, os servidores de *e-mail* funcionam da seguinte maneira:

- Cada servidor possui uma lista de contas de indivíduos ou endereços de grupos de indivíduos que acessam o serviço.
- Cada conta possui sua respectiva área de armazenamento (*mailbox*) no servidor.
- Quando um servidor recebe uma mensagem, ele identifica a conta do destinatário e armazena a mensagem em sua respectiva área.
- A área de armazenamento acumula uma série de mensagens recebidas até quando o dono da conta acesse sua aplicação cliente de *e-mail* para conectar-se ao servidor e receber as mensagens.

2.3 Anti-Spam Barracuda

Um dos objetivos deste trabalho de dissertação é a comparação do desempenho do SAS na classificação de *e-mails* com o do sistema comercial Barracuda (Bravo Tecnologia, 2013), utilizado na Universidade Federal de Itajubá.

O anti-*spam* Barracuda é uma solução integrada de *hardware* e *software*. Faz uso de 12 filtros para detecção de *e-mails* com conteúdo *spam*, de vírus, de *phising* e de *spyware*.

A Universidade Federal de Itajubá empregou um *cluster* de dois servidores r200 Dell, cada qual com um processador Intel Xeon de 4 núcleos de 2.0Ghz para a execução do Barracuda. De 2005 a 2013, último ano de utilização do Barracuda pela Universidade, a base do Barracuda é treinada com uma amostragem de 300 *e-mails*/dia. O *cluster* recebeu uma média de 120 mil *e-mails*/dia, dos quais 80% foram classificados como *spam* ou vírus.

O Barracuda classifica os *e-mails* em *ham* e *spam* de acordo com alguns critérios. Dentre estes, incluem-se a reputação de usuários e servidores de *e-mails* (listas negras e brancas), análise de conteúdo, dentre outros. Para fins de comparação com o SAS, na base de dados coletada (seção 4.4), encontram-se apenas *e-mails* que foram classificados pelo Barracuda por análise de conteúdo.

2.4 Codificação HTML

A mensagem do *e-mail* pode ser constituída de *plain text* (texto puro), codificação HTML (*HyperText Markup Language*), ou de ambos. *E-mails* com *tags* HTML são compostos pelo texto com comandos de formatação que são interpretados pela aplicação cliente de *e-mail* do usuário e apresentados em seu computador.

A opção de envio de *e-mail* nos formatos citados anteriormente (*plain text* ou HTML) geralmente é fornecida pela própria aplicação cliente de *e-mail*. Desta forma, o emissor da mensagem tem total controle sobre a mensagem que está sendo enviada.

2.4.1 Tags HTML

Na estrutura de um *e-mail* em formato HTML, os comandos usados para formatação são chamados de *tags*. Cada *tag* possui dois símbolos: o sinal "<"("menor que") e o sinal ">"("maior que").

O comando é digitado entre esses símbolos, como, por exemplo, *head* (cabeçalho) ou *body* (corpo), e pode estar tanto em letras maiúsculas quanto minúsculas. *Tags* podem ser independentes, como, por exemplo, a *tag*
, ou podem vir em pares, como, por exemplo, ... Abaixo estão apresentados alguns exemplos de *tags* HTML:

- Negrito (*Bold*): texto ou texto. Deixa o texto em negrito.
- Itálico (*Italic*): texto ou texto. Deixa o texto em itálico.
- Tachado (*Strikethrough*): <strike>texto</strike> ou <s>texto</s> ou texto. Deixa o texto riscado.

2.4.2 Estrutura básica HTML

Todo arquivo HTML obrigatoriamente contém *tags* HTML que o identificam como tal. Assim, *e-mails* em HTML possuem uma estrutura fixa e devem conter obrigatoriamente as seguintes *tags*:

- <html> e </html> - Determinam início e fim da mensagem HTML. <html> diz à aplicação cliente de *e-mail* para iniciar um novo documento HTML cujo conteúdo se encontra definido após esta *tag* e a *tag* </html>.
- <body> e </body> - Define o que a aplicação cliente de *e-mail* deve apresentar graficamente. Todos os arquivos, textos, tabelas, sons e vídeos devem estar entre estes elementos.

Deve-se ressaltar que a linguagem de marcação HTML respeita uma hierarquia, onde o primeiro elemento a ser aberto (no caso, <html>) é sempre o último a ser finalizado.

2.5 Codificação de Caracteres

Ao visualizar uma mensagem, a acentuação pode aparecer de forma confusa e caracteres não identificados podem estar presentes. É comum que letras com acentos e ç apareçam como *?* ou como outros símbolos. Este fato ocorre porque a aplicação cliente não utilizou o padrão correto de descodificação, utilizado na codificação dos caracteres.

Existem padrões empregados na codificação dos caracteres presentes nos *e-mails*, como, por exemplo, ISO 8859-1, UTF-8, dentre outros. Entretanto, se a mensagem não informar a codificação empregada, a aplicação cliente pode formatar incorretamente o texto. Dois dos mais importantes padrões de codificação são:

- ISO 8859-1 (*International Standardization Organization*): É o padrão ocidental, utilizado também no Brasil. Cada caractere só possui 1 *byte* (8 bits), o que permite um máximo de 256 caracteres no padrão.
- UTF-8: Padrão mundial, que pode ser usado em quase todos os idiomas. Cada caractere possui 2 *bytes* (16 bits), o que permite um valor máximo (65.536) de caracteres maior que o padrão anterior.

2.6 Arquivos XML

XML, do inglês *eXtensible Markup Language*, é uma linguagem de marcação recomendada pela W3C para a criação de documentos com dados organizados hierarquicamente, tais como: textos, banco de dados ou desenhos vetoriais. A linguagem XML é classificada como extensível porque permite definir novas *tags* de marcação.

2.6.1 Principais Características

Esta linguagem traz uma sintaxe básica que, por ser padronizada, permite um fácil compartilhamento das informações entre diferentes computadores e aplicações. Quando combinada com outros padrões, permite definir o conteúdo de um documento separadamente de seu formato, facilitando a reutilização do código em outras aplicações para diferentes propósitos.

Portanto, uma de suas principais características é sua portabilidade. Por exemplo, arquivos XML podem ser importados e exportados por diferentes sistemas gerenciadores de banco de dados. A Figura 2 contém um exemplo de texto formatado com *tags* XML.

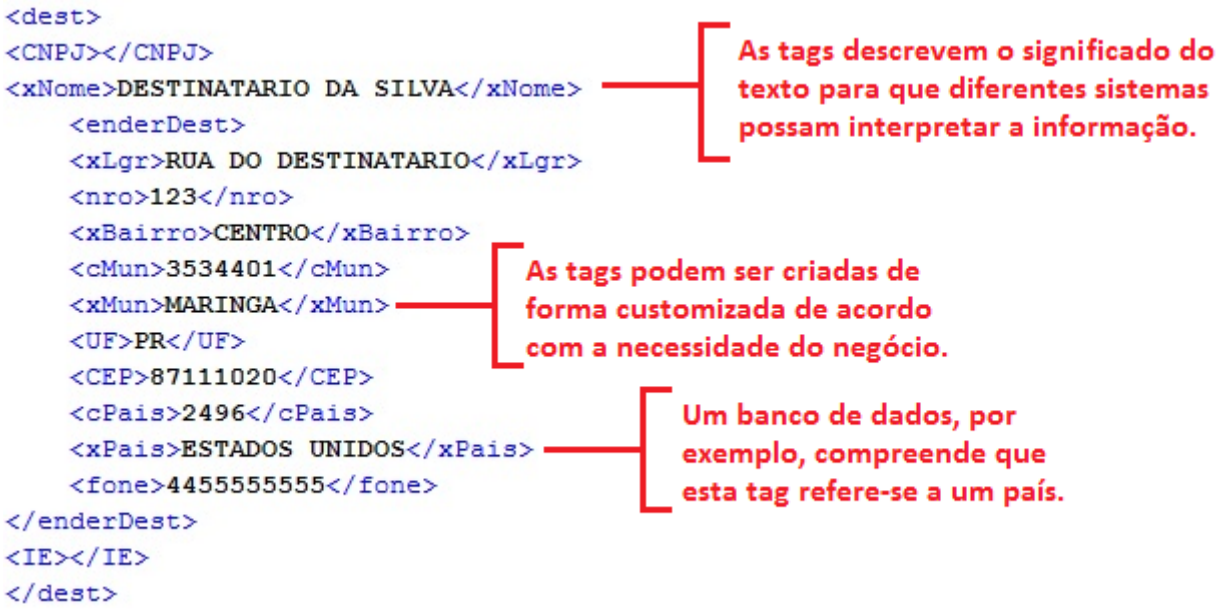

Figura 2: *Exemplo de Código XML*

2.7 Técnicas de Ofuscamento

Como visto na seção 1.5, à medida que as técnicas empregadas nos filtros anti-*spam* evoluem, os *spammers* aprimoram e adaptam suas técnicas para que seus *e-mails spam* não sejam identificados (COURNANE *et al.*, 2004).

As técnicas de ofuscamento adotadas por *spammers* variam desde a maneira como o *e-mail spam* é enviado, até a forma com a qual ele se apresenta ao usuário final. É importante observar um constante aprimoramento de técnicas por ambas as partes (*spammers* e filtros) (COURNANE *et al.*, 2004).

2.7.1 Ofuscamento no envio de E-mails

Devido ao avanço das técnicas estáticas, apresentadas na seção 1.5.1, para filtrar *e-mails spam*, *spammers* passaram a empregar mecanismos de ofuscamento para confundir tais métodos estáticos (ANTISPAM, 2013). Por exemplo:

- *Spam zombies*: São computadores infectados por códigos maliciosos que os transformam em servidores de *e-mail* para envio de *spam*. A maior parte dos códigos maliciosos propaga-se por *e-mail*.
- *Bulk mailing*: São programas maliciosos que se apropriam de máquinas mal configuradas para enviar *e-mails* em massa sem o conhecimento do usuário.
- Servidores de *spam*: São servidores de *e-mail* criados exclusivamente para envio de *spams*. São configurados de forma a ofuscar a origem das mensagens para burlar os filtros de *e-mail*.
- Vírus por *e-mail*: São programas executáveis que chegam anexados aos *e-mails*. Os usuários que os instalam têm suas máquinas infectadas e, geralmente, passam a reproduzir estes *e-mails* para suas listas de contatos. Uma variação desta abordagem ocorre quando, ao invés de um anexo presente, o *e-mail* apresenta um *link* externo para um *site* infectado.

2.7.2 Ofuscamento em Mensagens

Outra forma encontrada pelos *spammers* para burlar a filtragem de *e-mail* consiste em manipular seu conteúdo. Eles empregam técnicas que visam fazer com que o conteúdo *spam* se pareça com uma mensagem *ham*. Este ofuscamento ocorre no corpo do *e-mail* e faz com que os filtros, que utilizam usualmente métodos probabilísticos de seleção de

características, não consigam classificar corretamente a mensagem como *ham* ou *spam* (COURNAME *et al.*, 2004).

A seguir estão apresentadas algumas das técnicas empregadas no corpo das mensagens para ofuscamento de conteúdo *spam*:

- *Blank* HTML: Nesta técnica, o conteúdo *spam* não está presente no texto e sim em imagens incluídas no corpo do *e-mail*. Assim, os filtros convencionais não conseguem analisar os textos presentes nas imagens e, por consequência, não detectam o *spam*.
- Texto invisível: Consiste em esconder um texto *ham* em um *e-mail spam*, para que a mensagem pareça legítima. Para tal, o *spammer* emprega *tags* HTML que ocultam ao usuário o texto legítimo e exibem somente o conteúdo *spam*. Como muitos filtros classificam *e-mails* através do percentual de palavras consideradas válidas e suspeitas que estes contêm, os textos *ham* confundem os filtros fazendo com que classifiquem os *e-mails* incorretamente. Uma das formas de ocultar o texto *ham* consiste em formatá-lo na mesma cor do fundo da tela (*background*) do monitor do computador (geralmente branco).
- *Tags* HTML de comentário ou *tags* vazias: Visa dividir palavras em textos *spam* com *tags* HTML de comentário ou vazias. O comentário HTML e a *tag* vazia não são processados pela aplicação cliente de *e-mail*, logo não são exibidos. Assim, as palavras são mostradas intactas, sem divisões, para o usuário. Como dito anteriormente, os filtros que trabalham considerando o percentual de palavras válidas e suspeitas não conseguem identificar estas "novas" palavras divididas e, portanto, não as contabilizam como suspeitas. Tem-se, como exemplo, <html> <body> dinh   eiro </body> </html>.
- *Vertical Slicing*: Consiste em utilizar *tags* HTML para criar tabelas com bordas transparentes, para esconder palavras com alto potencial de suspeição. Como o sistema anti-*spam* não identifica tais palavras, separadas em células de uma tabela, o *e-mail* acaba sendo classificado incorretamente como legítimo. A Figura 3 apresenta um exemplo desta técnica.
- *Mime segments*: Trata-se de mesclar textos do tipo *plain text* (com texto válido) e textos formatados em HTML com conteúdo *spam*. Apesar da aplicação cliente de *e-mail* só exibir para o usuário a codificação HTML com conteúdo *spam*, o sistema anti-*spam* classifica o *e-mail* como válido.

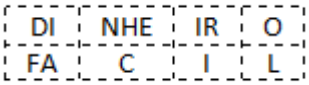

Figura 3: *Exemplo de tabela invisível em HTML*

- Acentuação incorreta: Os *spammers* acentuam o texto de forma errada com a finalidade de enganar os filtros que não retiram as acentuações das palavras.

- Espaçamento de letras: Uma forma empregada pelos *spammers* para ofuscar as mensagens é a utilização de espaços entre as letras das palavras. Assim, o sistema anti-*spam* não identifica as palavras, mas só letras soltas. Variantes desta forma incluem a utilização de pontos ou caracteres especiais para separarem as letras. Por exemplo: d i n h e i r o, d.i.n.h.e.i.r.o ou d*i*n*h*e*i*r*o.

- Codificação URL: Ao invés de empregar o endereço de um *site* claramente reconhecido como *spam*, os *spammers* substituem o nome do *link* por seu endereço IP. Por exemplo, http://www.google.com, substituído por 190.98.170.104.

2.8 Método de Seleção de Características

A quantidade de características (palavras e *tags* HTML) diferentes presente nos *e-mails* aumenta a complexidade da classificação. Maior quantidade de atributos significa maior dimensionalidade dos vetores de características que representam os *e-mails* e, por conseguinte, significa maior tempo computacional gasto pelo sistema anti-*spam* para classificação dos *e-mails* (JOACHIMS, 1998). Para contornar o problema da dimensionalidade, aplicam-se métodos estatísticos de seleção de características antes da etapa de classificação, para selecionar as características mais relevantes, reduzindo-se, assim, a dimensionalidade e o custo computacional (SEBASTIANI, 2002).

Métodos de seleção de características selecionam as características mais relevantes, que serão representadas por coordenadas de vetores. Cada vetor, representando um determinado *e-mail*, é submetido à entrada da rede neural MLP (seção 2.8) para ser classificado como *ham* ou *spam*.

Neste trabalho, foram avaliados três métodos de seleção de características — X^2

Statistic (CHI), *Frequency Distribution* (DF) e *Mutual Information* (MI) —, escolhidos de acordo com o desempenho alcançado em outros trabalhos reportados na literatura (JOACHIMS, 1998).

2.8.1 X^2 Statistic

Este método mede o grau de dependência entre um termo t e uma categoria c, segundo a equação:

$$X^2(t_k, c_i) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)} \tag{2.1}$$

A representa o número de documentos em que t e c ocorrem; B o número de documentos em que t ocorre sem c; C o número de documentos em que c ocorre sem t; D o número de documentos em que nem t nem c ocorrem; e N representa o número total de documentos que compõem o conjunto de treino (SEBASTIANI, 2002).

Esta equação resulta em zero se t e c são independentes. Calcula-se, para cada categoria, o valor do X^2 entre cada termo único do corpo de treinamento e uma categoria. Para obter a importância global do termo, combinam-se os valores específicos de um termo nas diversas categorias em dois possíveis cálculos (equações 2.2 e 2.3).

$$X^2_{max}(t_k) = \max_{i=1}[X^2(t_k, c_i)] \tag{2.2}$$

$$X^2(t_k) = \sum_{i=1}^{|c|} P(c_i) X^2(t_k, c_i) \tag{2.3}$$

A computação dessa técnica tem complexidade quadrática, similar à *Mutual Information*. A maior diferença entre MI e X^2 é que a segunda tem o valor normalizado, então os valores são comparáveis entre os termos de uma mesma categoria. Entretanto, essa normalização é ruim para termos de baixa frequência.

2.8.2 Frequency Distribution

Uma distribuição de frequências agrupa um grande número de dados em uma tabela, de modo que 100, 200, 500 ou um número qualquer de valores possa ser representado em poucas linhas. É uma série estatística específica em que os dados encontram-se dispostos em classes ou categorias juntamente com suas frequências correspondentes. Com isso, pode-se resumir e visualizar um conjunto de dados sem precisar levar em conta os valores individuais (SEBASTIANI, 2002).

A equação utilizada para distribuição de frequência (DF) tem por objetivo escolher atributos de forma a maximizar os índices de DF. Ela mede o grau de ocorrência de um elemento W em um conjunto C. Se W é uma característica, a distribuição de frequência é (equação 2.4):

$$DF(w) = \frac{N(w \in [spam, ham])}{T} \tag{2.4}$$

2.8.3 Mutual Information

Dado um termo t_k e uma categoria c_i, considere A o número de vezes que t_k e c_i ocorrem; B o número de vezes que t_k ocorre sem c_i; C o número de vezes que c_i ocorre sem t_k; e N o número total de documentos de treinamento (KUSHMERICK e THOMAS, 2003). A medida de informação mútua é definida como (equação 2.5):

$$MI(t_k, c_i) = \frac{P(t_k \wedge c_i)}{P(t_k)P(c_i)} \tag{2.5}$$

Estima-se a medida da informação mútua usando-se a equação 2.6:

$$MI(t_k, c_i) \approx \frac{\log A \times N}{(A + C)(A + B)} \tag{2.6}$$

A medida MI (t_i, c_i) tem o valor de zero caso t_i e c_i sejam independentes. Para medir

a importância de um termo globalmente, combinam-se as pontuações específicas de um termo para cada categoria em duas formas alternativas (equações 2.7 e 2.8):

$$MI_{avg}(t_k) = \sum_{i=1}^{|c|} P(c_i) MI(t_k, c_i) \tag{2.7}$$

$$MI_{max}(t_k) = \max_{i=1}[MI(t_k, c_i)] \tag{2.8}$$

Desta forma, para cada termo, é calculada a informação mútua e são removidos do *corpus* os termos que possuírem um valor inferior a um limiar pré determinado (KUSHMERICK e THOMAS, 2003).

2.9 Redes Neurais Artificiais

Uma rede neural artificial é um modelo matemático. Foi inspirado no cérebro e na maneira como ele processa a informação. O elemento principal deste modelo é sua estrutura para processamento de informação. A estrutura é composta por um número de elementos processadores interconectados (neurônios) que trabalham em conjunto para resolver problemas (HAYKIN, 2000).

Estas redes, similares aos modelos biológicos, aprendem com treinamento. São configuradas, através do treinamento, para realizar tarefas específicas, como reconhecimento de padrões e classificação de dados. Assim, como nos sistemas biológicos, o treinamento (ou aprendizado) consiste no ajuste das conexões sinápticas existentes entre os neurônios artificiais (BRAGA *et al.*, 2007).

2.9.1 Vantagens das Redes Neurais

Redes neurais, por sua capacidade de lidar com dados complexos e imprecisos, podem ser usadas para extrair padrões e detectar tendências, padrões e tendências estes que são complexos para serem extraídos ou detectados por humanos ou por outras modelos

computacionais. Uma rede treinada pode ser considerada como um *expert* no domínio em que ela atua. Pode ser utilizada para fornecer previsões, dando respostas para novas situações e simulações.

Outras vantagens destes modelos incluem (BISHOP, 1992):

- Treinamento adaptativo: Habilidade de aprender como fazer e agir de acordo com os dados utilizados no treinamento inicial
- Organização própria: Podem criar seu próprio método de representação para a informação recebida durante o treinamento.
- Operação em tempo real: Sistemas neurais podem trabalhar em paralelo, *hardwares* especiais são criados e projetados para tirar proveito desta capacidade.
- Tolerância a falhas através de codificação de informação redundante: Destruição parcial da rede leva à correspondente degradação de desempenho. Entretanto, algumas capacidades da rede podem ser mantidas mesmo com dano crítico a ela.

2.9.2 Neurônios

O neurônio artificial básico é um componente com várias entradas e uma saída. Ele possui dois modos de operação: treinamento e utilização (Figura 4). No modo de treinamento, ele pode ser treinado para responder positiva ou negativamente de acordo com os padrões de entrada. Já em modo de utilização, quando um padrão de entrada previamente treinado é detectado, a respectiva saída é emitida. Caso a entrada não tenha feito parte do treinamento, mesmo assim o neurônio produz uma resposta, correta ou incorreta, seguindo as regras de ativação configuradas pelo treinamento (HAYKIN, 2000).

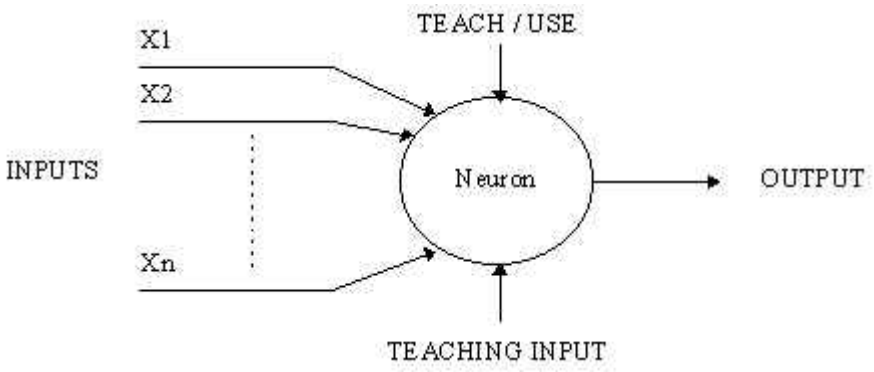

Figura 4: *Modelo de neurônio*

Fonte: http://www.doc.ic.ac.uk/nd/journal/vol4/cs11/report.simple_neuron.jpg

2.9.3 Regras de Ativação

Representam um importante conceito na área de redes neurais devido à alta flexibilidade. Estas regras determinam como e quando um neurônio deve emitir uma resposta de acordo com o padrão recebido em sua entrada, abrangendo todas as possíveis entradas e não somente as que foram previamente treinadas (BRAGA *et al.*, 2007).

2.9.4 Perceptron (Neurônios MCP)

O neurônio artificial perceptron possui em suas entradas pesos que influenciam o resultado produzido (HAYKIN, 2000).

Um determinado valor numérico (peso) é inserido antes de cada entrada dos dados no neurônio. Caso a soma de todos os valores ponderados (pelos pesos) ultrapasse um valor pré-determinado, o neurônio então é ativado (Figura 5). Estes neurônios têm a capacidade de se adaptar a cada situação particular, através do ajuste dos pesos realizado no treinamento (HAYKIN, 2000).

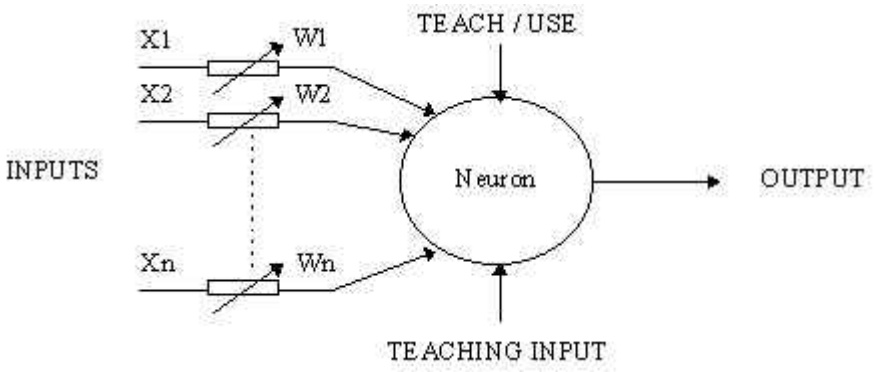

Figura 5: *Representa o somatório X1W1 + X2W2 + X3W3 + ... > T*

Fonte: http://www.doc.ic.ac.uk/nd/journal/vol4/cs11/report.mcp_neuron.jpg

2.9.5 MLP e Backpropagation

O *perceptron* multicamadas (MLP) consiste em uma rede neural com uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída, todas compostas por neurônios artificiais. O sinal de entrada propaga-se para frente, através da rede, camada por camada (BISHOP, 1992). A Figura 6 apresenta um MLP.

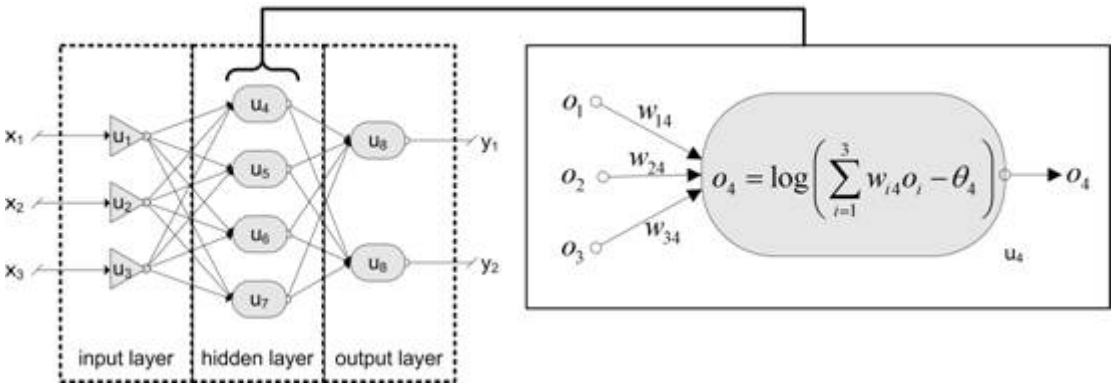

Figura 6: *Topologia MLP*

Fonte: http://www.neural-forecasting.com/mlp_neural_nets.htm

A rede MLP, através do seu treinamento de forma supervisionada com um algoritmo de retro propagação (*backpropagation*), tem sido aplicada com sucesso na resolução de problemas de alta complexidade. O algoritmo de retro propagação baseia-se na regra de aprendizagem por correção de erro. Como tal, pode ser visto como uma generalização de

um algoritmo de filtragem adaptativa chamado mínimo quadrado médio (LMS) (BISHOP, 1995).

A aprendizagem por retro propagação de erro consiste em dois passos: um passo para frente, a propagação, e um passo para trás, a retro propagação. No passo para frente, um padrão de atividade (vetor de entrada) é aplicado aos neurônios da rede e seu efeito se propaga através da rede, camada por camada. Finalmente, um padrão de saída é produzido como resposta real da rede. Durante o passo de propagação, os pesos sinápticos da rede são todos fixos. Durante o passo para trás, por outro lado, os pesos sinápticos são todos ajustados de acordo com uma regra de correção de erro. Especificadamente, o padrão de saída é comparado ao padrão desejado (alvo) para produzir um sinal de erro (BISHOP, 1995).

Este sinal de erro é, então, propagado para trás (*backpropagation*), através da rede, contra a direção das conexões sinápticas. Os pesos sinápticos são ajustados para fazer com que a resposta real da rede se mova estatisticamente para mais perto da resposta desejada (BISHOP, 1995).

2.10 Considerações Finais

Neste capítulo foi apresentada a teoria que fundamenta a pesquisa realizada neste trabalho. O SAS desenvolvido possui módulos para identificar diversas técnicas de ofuscamento, módulos que implementam os três métodos de seleção de características e um módulo que implementa um modelo neural MLP, todos apresentados neste capítulo. Estes módulos podem ser ativados ou desativados, através de um arquivo XML de configuração, pelo usuário.

No próximo capítulo, é apresentada uma revisão da literatura relacionada à área deste trabalho, apresentando os modelos e técnicas empregados.

3 Revisão Bibliográfica

3.1 Considerações Iniciais

Este capítulo apresenta uma revisão da literatura relacionada à área de sistemas anti-*spam*, apresentando os modelos e técnicas empregados.

3.1.1 Trabalhos Revisados

Silva, Moita e Almeida (SILVA *et al.*, 2010) propuseram o uso do mapa auto-organizável (SOM) como filtro anti-*spam*, bem como o uso de pré-processamento e seleção de características dos *e-mails*. Através de técnicas de detecção de padrões, a SOM consegue criar um mapa topológico organizado com as regiões correspondentes aos *e-mails hams* e *spams*. A base de *e-mails* empregada nos estudos foi coletada de um dos subdomínios do CEFET-MG. Compreende 12.687 *e-mails* (divididos em 8.867 *hams* e 4.020 *spams*) pertencentes a usuários com diferentes perfis de utilização de mensagens eletrônicas. Para treinamento, validação e teste da SOM, a base de dados foi distribuída em três subconjuntos sendo 60% para treinamento, 20% para validação e 20% para teste. Utilizando vetores de entrada com 25 características e o método de distribuição de frequência, a SOM alcançou seu melhor resultado, com 96,25% de acerto na classificação.

Chuan, Xianliang, Mengshu e Xu (CHUAN *et al.*, 2005) realizaram experimentos com três modelos de filtros anti-*spam*. O primeiro foi baseado no classificador *Naïve Bayesian*, o segundo em um classificador neural MLP e o terceiro em um *Learning Vector Quantization* (LQV). Com a utilização da base de *e-mails Spam Assassin*, o modelo *Naïve Bayesian* obteve 86,48% de precisão na classificação de *spam*, o classificador MLP obteve

91,26% e o LVQ obteve 93,58% de precisão na classificação de *spam*.

Sabri, Mohammads, Al-Shargabi e Hamdeh (SABRI *et al.*, 2010) propõem um sistema anti-*spam* baseado em um modelo neural artificial com camadas de entrada adaptáveis. O sistema compara os atributos presentes nos *e-mails* com os cadastrados nas camadas de entrada. Novos atributos encontrados em *e-mails spam* formam novas camadas de entrada da rede neural. Utilizando 682 *spams* e 3.435 *hams* da base *Spam Assassin* e o modelo neural com 900 unidades nas camadas de entrada, obtiveram, como melhor resultado, 99,14% de precisão na classificação de *e-mails spam*, 0,09% de falsos positivos e 5,2% de falsos negativos.

Clark, Koprinska e Poon (CLARK *et al.*, 2003) desenvolveram uma rede perceptron multicamadas treinada com *backpropagation* para classificar *e-mails* por assunto e por *spam* através da seleção das palavras mais frequentes presentes no corpo de cada mensagem. Para testes foram empregadas cinco amostras totalizando 4.274 *e-mails* divididos em 1.044 *spams* e 3.230 *hams*. Na classificação por assunto obtiveram como melhor resultado uma amostra com 92,34% de acerto e como pior resultado uma amostra com 68.54% de acerto. Atribuíram esta diferença aos diferentes temas das mensagens analisadas. Na classificação por *spam*, obtiveram 95,62% de precisão no melhor resultado.

Shi, Wang, Ma, Weng e Qiao (SHI *et al.*, 2012) empregaram um sistema de árvores de decisão combinadas que verifica e classifica o conteúdo das mensagens de acordo com 58 principais características presentes em *spams*, previamente retiradas da base SPAM *E-mail Database*. Com uma base de dados de 4.601 *e-mails*, sendo 1.813 *spams* e 2.788 *hams*, obtiveram 94.6% de precisão com o sistema proposto, contra 90.6% da ANN, 83.5% da SVM e 79.2% da *Naïve Bayesian*. Nesta proposta não foram analisadas a capacidade de adaptação do sistema frente a novas características de *spams* e o uso de mensagens eletrônicas em outro idioma além do inglês.

Ma, Tran e Sharma (MA *et al.*, 2009) propuseram o uso de seleção negativa, conceito proveniente de sistemas de imunidade artificial, para detectar novas formas de *e-mail spam*. Eles não fazem uso de nenhum conhecimento prévio sobre *spams* porque consideram como indesejada qualquer nova mensagem. O programa desenvolvido codifica cada nova mensagem em uma *string* hexadecimal de 256-*bits*. A classificação é realizada através dos padrões resultantes da comparação das *strings* hexadecimais codificadas. Eles

empregaram 20,000 *e-mails* (4,890 *hams* e 14,489 *spams*) da base TREC07 e obtiveram 78% de precisão como taxa média de detecção de *spams*. O diferencial deste trabalho é seu bom desempenho frente a técnicas de ofuscamento desconhecidas.

Li e Huang (LI *et al.*, 2012) utilizaram uma rede neural adaptada, treinada com *backpropagation*, para detecção de *spam*. A adaptação feita pelos autores na rede neural consiste no uso de uma variável estatística, que avalia cada etapa de aprendizagem, aumentando a velocidade de treinamento da rede. Utilizam algoritmos de decomposição para reduzir a complexidade dos *e-mails*, facilitando, deste modo, a extração de suas características. A base de dados possui 2.893 *e-mails*, 2.412 *hams* e 481 *spams* classificados manualmente, retirados do repositório Ling-*Spam*, e 5.238 *e-mails* (2.934 *hams* e 2.304 *spams*) coletados de dois repositórios particulares dos autores. O melhor resultado de classificação atingiu 97,5% de precisão sobre os *e-mails*.

Stuart, Cha e Tappert (STUART *et al.*, 2004) compararam o desempenho, na classificação de *e-mails*, de uma rede neural perceptron multicamadas com um modelo *Naïve Bayesian*. Eles utilizaram uma base de 1.654 *e-mails*, sendo 800 legítimos e 854 *spams*. A rede neural alcançou 92,45% de precisão na detecção de *spams* e 91,32% na de *hams*, contra, respectivamente, 99% e 96,2% do modelo *Naïve Bayesian*. Os autores atribuíram o baixo desempenho da rede neural à fraca implementação da rede neural utilizada (com apenas 12 neurônios) e ao baixo número de *e-mails* presentes na base. Ressaltaram o fato de que o modelo *Naïve Bayesian* alcança alta precisão apenas quando os *e-mails* apresentam padrões já conhecidos.

Goweder, Rashed, Elbekaie e Husien (GOWEDER *et al.*, 2008) propuseram um filtro anti-*spam* baseado em um classificador neural MLP treinado com Algoritmo Genético (GA). Escolheram o Algoritmo Genético como método de treinamento devido aos processos de interações em paralelo entre diferentes genes da população de possíveis candidatos. Eles usaram *e-mails* de três bases diferentes — 1000 *e-mails* da base *Spam Assassin* (630 *spams* e 370 *hams*), 1000 *e-mails* da base TREC 2005 (630 *spams* e 370 *hams*) e 72 *e-mails* da base *Arabic Corpus* (56 *spams* e 16 *hams*). Os *e-mails* das bases foram pré processados, para remoção de características irrelevantes, para normalização das características e geração de *tokens*. Obtiveram, como resultado, 94% de taxa de detecção de *spams* e 89% de taxa de detecção de *hams*. Os melhores resultados foram obtidos sobre *e-mails* em inglês, devido ao seu Pré-Processamento ser mais efetivo.

Androutsopoulos, Georgios, Paliouras, Karkaletsis, Spyropoulos e Stamatopoulos (ANDROUTSOPOULOS *et al.*, 2003) desenvolveram um sistema que compara o conteúdo dos novos *e-mails* com o de *e-mails* de uma base armazenada previamente em memória. Os resultados obtidos com o sistema foram comparados aos obtidos por um modelo *Naïve Bayesian*. A base de dados utilizada possui 2.893 *e-mails*, sendo 2.412 *hams* e 481 *spams*, retirados da base Ling-*spam*. O sistema armazena, em memória, os *e-mails spam* do conjunto de treinamento e calcula o grau de similaridade entre cada novo *e-mail* e os *e-mails* do conjunto de treinamento. O sistema alcançou 97,39% de precisão na classificação de *spams* contra 82,35% alcançado pelo modelo *Naïve Bayesian*.

Tak e Tapaswi (TAK & TAPASWI, 2010) propuseram um sistema que envolve um sistema gerenciador de banco de dados e uma rede neural para classificação de *e-mails*. Cada novo *e-mail* passa por passos que visam classificá-lo como *spam* ou *ham*. Cada passo analisa um atributo do *e-mail*, tal como, o assunto do *e-mail*, o país de origem, a reputação do emissor. O passo final compara as características do texto do *e-mail* com características suspeitas, com potencial *spam*, armazenadas em uma base previamente criada. Caso o *e-mail* não seja classificado em nenhum dos passos, seu conteúdo passa por uma rede neural, que o classifica de acordo com os dados levantados previamente. A base de dados, composta por 69.414 *e-mails* (26.451 *spams* e 42.963 *hams*), foi coletada, durante 4 meses, de caixas de entrada de *e-mails* de um grupo de usuários. O sistema obteve 98,17% de taxa de detecção de *spams* e 0,12% na taxa de falsos positivos. Segundo os autores, devido a execução de várias *queries* no banco de dados para comparação de atributos, o sistema proposto torna-se inviável, exigindo sistemas computacionais de grande desempenho e com grande capacidade de memória.

3.2 Considerações Finais

Dois artigos fazem uma boa revisão das técnicas empregadas pelos *spammers*, bem como das propostas existentes de sistemas anti-*spam*.

Cournane *et al.* (COURNANE *et al.*, 2004) apresentaram vários problemas oriundos dos *spams*, como prejuízos causados por seu tráfego, fraudes, bem como abordam mecanismos para combatê-los. Listam algumas das ferramentas existentes para coleta de endereços de *e-mails*. Apresentam, detalhadamente, algumas das mais importantes

técnicas de ofuscamento de conteúdo empregadas por *spammers*. Abordam, por fim, os aspectos jurídicos relacionados ao envio de *spam*.

Subramaniam, Jalab e Taqa (SUBRAMANIAM *et al.*, 2010) fazem uma revisão dos modelos para detecção de *spam*, tais como *Naïve Bayesian*, máquinas de vetores de suporte e redes neurais, evidenciando suas qualidades e restrições. Tal como o trabalho de Cournane *et al.*, apresentam prejuízos decorrentes da circulação de *spams* na *Internet*, bem como os aspectos jurídicos ligados ao assunto. Afirmam que o emprego de técnicas de pré-processamento auxilia os modelos para detecção de *spam*.

O próximo capítulo detalha o modelo anti-*spam* desenvolvido neste trabalho de dissertação e os resultados obtidos pelo mesmo.

4 Modelo, Resultados e Análises

4.1 Considerações Iniciais

Este capítulo apresenta o modelo do sistema anti-*spam* (SAS) desenvolvido, a base de dados utilizada, os resultados obtidos pelo SAS na classificação dos *e-mails* e, por fim, compara-os com os obtidos pelo sistema anti-*spam* comercial Barracuda.

4.2 SAS

Como descrito anteriormente no Capítulo 1, a nova versão do SAS é um sistema anti-*spam* composto por três módulos interligados — Pré-Processamento, Seleção de Características e Classificador — e desenvolvido em linguagem Java. Os módulos são descritos nas subseções a seguir, de acordo com sua ordem de execução. A Figura 7 ilustra como era a versão antiga do SAS e a Figura 8 mostra como ficou a nova versão unificada.

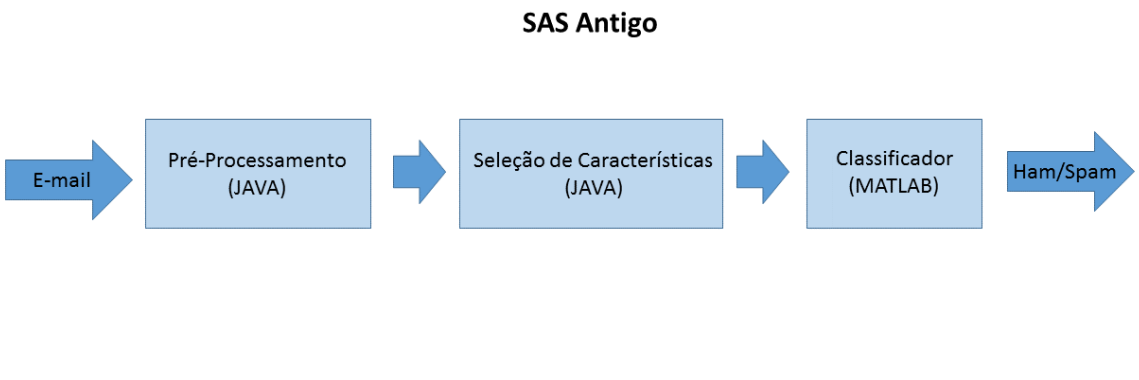

Figura 7: *Versão antiga do SAS realizada no GPESC*

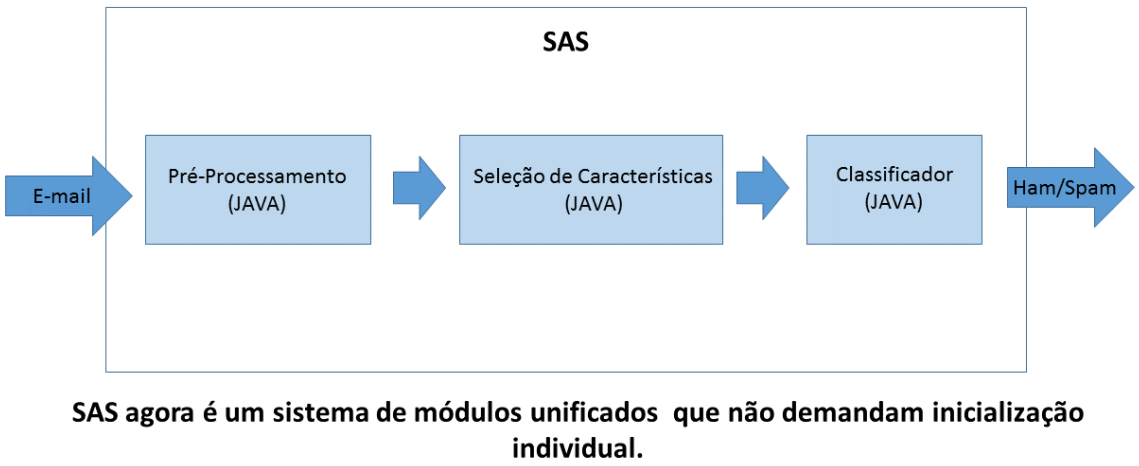

Figura 8: *Nova versão do SAS criada*

4.2.1 Módulo de Pré-Processamento

O módulo de pré-processamento tem por objetivo reduzir a complexidade dos textos dos *e-mails* e identificar técnicas de ofuscamento presentes nos *e-mails*. O processamento realizado pelo módulo de pré-processamento é dividido em diversos passos:

4.2.1.1 Passo 1: Análise dos anexos e da formatação do texto do e-mail

No primeiro passo, caso o *e-mail* possua anexos, estes são descartados e no corpo do *e-mail* é adicionada a *tag* "!_ANEXO_<extensão>_<nome>". Por exemplo, se o *e-mail* possuir os anexos "AAA.pdf" e "BBB.exe", estes são removidos e no corpo do *e-mail* são acrescentadas as *tags* "!_ANEXO_pdf_AAA" e "!_ANEXO_exe_BBB", respectivamente.

No primeiro passo, é, igualmente, verificada a formatação do texto do *e-mail*. Se o *e-mail* não possui *tags* HTML, é encaminhado para o terceiro passo deste módulo. Caso contrário, é encaminhado para o segundo passo.

4.2.1.2 Passo 2: Interpretação de tags HTML

No segundo passo, as *tags* HTML do *e-mail* são analisadas de acordo com cinco categorias:

Primeira Categoria: Nesta categoria, encontram-se *tags* de formatação de texto, de cabeçalho de página, as *tags style, title, marquee* e similares, bem como *tags* de comentários que não alteram as cores das fontes ou do plano de fundo do *e-mail*. *Tags* nesta categoria são descartadas.

Segunda Categoria: Nesta categoria, encontram-se *tags* de formatação que alteram cores de fontes de texto e de planos de fundo dos *e-mails*. Caso alguma *tag*, tal como, "*font color*" ou "*background color*", seja identificada, é verificado se ela implementa a técnica de ofuscamento conhecida como "texto invisível" (descrita na seção 2.5.2). Caso implemente a técnica de "texto invisível", a *tag* e seu conteúdo são substituídos por uma nova *tag* "!_invisible_text".

Terceira Categoria: *Tags* nesta categoria têm seus atributos removidos. A própria *tag* é substituída por uma nova *tag*. Nesta categoria, incluem-se as *tags abbr, var, tr, td, textarea, p, option, ol, acronym, b, body, br, caption, col, del, frame, h1 h6, head, hr, html, i, ins, label* e *li*. Um exemplo de processamento desta categoria é o caso da *tag* "*<head> Conteúdo Interno</head>*" que é substituída pelo indicador "!_in_head *Conteúdo Interno*".

Quarta Categoria: Nesta categoria, inclui-se apenas a *tag* "*table*". Quando identificada no corpo do *e-mail*, é verificado se ela implementa a técnica de ofuscamento conhecida como "*vertical slicing*" (descrita em 2.5.2). O conteúdo da tabela é sempre convertido em texto e a *tag* "*table*" é substituída por uma nova *tag* "!_TABLE conteúdo_da_tabela". Além disto, se houver colunas vazias na tabela, estas são substituídas pela *tag* "!_BLANK-COLUMN".

Quinta Categoria: *Tags* nesta categoria têm seus conteúdos processados completamente. Nesta categoria, incluem-se as *tags input, map, img, font, form, button, base* e *a*.

4.2.1.3 Passo 3: Transformação em unidades lógicas (Tokenização)

No terceiro passo, o texto do *e-mail* é dividido em *tokens*. Esta divisão do texto é realizada segundo delimitadores definidos previamente. Os delimitadores pré-definidos, que separam as palavras, são espaços, tabulações, linhas, parágrafos e pontuação.

Algumas técnicas de ofuscamento empregam delimitadores para alterar palavras, como por exemplo, "d i n h e i r o". Neste caso, o *token* é mantido como "d i n h e i r o" e não quebrado nos *tokens* "d", "i", "n", "h", "e", "i", "r" e "o". Por fim, os acentos das letras contidas nos *tokens* são removidos e todas as letras são convertidas para o padrão minúsculo para uniformização.

4.2.1.4 Passo 4: Detecção de Padrões Spam

No quarto passo, os *tokens* são analisados para detectarem-se técnicas de ofuscamento e padrões de *spam*. A análise realizada sobre os *tokens* é descrita a seguir.

- *Tokens* com endereço de *e-mail* são substituídos pela *tag* "!_E-MAIL".
- *Tokens* com endereços de URL (independentemente da forma como são escritos os endereços) são substituídos pela *tag* "!_LINK".
- *Tokens* com caracteres especiais ou delimitadores entre letras são substituídos pela *tag* "!_HIDEWORDS". Por exemplo, "d-i-n-h-e-i-r-o" e "d i n h e i r o".
- *Tokens* com palavras extensas são substituídos pela *tag* "!_BIGTEXT".
- *Tokens* contendo sequências de números e caracteres no campo assunto (subject) do cabeçalho do *e-mail* são substituídos pela *tag* "!_NUMERO_SUBJECT".
- *Tokens* com conteúdo monetário e porcentagem são substituídos pelas *tags* "!_MONEY" e "!_PORCENTAGEM", respectivamente.

4.2.2 Módulo de Seleção de Características

O segundo módulo do SAS seleciona as características (*tokens*) mais relevantes do corpo do *e-mail* para compor o vetor de entrada da rede neural. O vetor de entrada representa, portanto, o *e-mail*. Para seleção das características, o SAS permite que sejam usados qualquer um dos três métodos estatísticos de seleção de características descritos na seção 2.6.

Cada coordenada do vetor de entrada representa a quantidade de ocorrências de uma determinada característica (*token*) no *e-mail*. Os vetores de entrada são normalizados no intervalo [0,1] de acordo com a característica com maior número de ocorrências.

A utilização de métodos de seleção de características é recomendável, pois reduzem significativamente a dimensão dos vetores de entrada e, por conseguinte, os tempos para treinamento e execução da rede neural. Os vetores de entrada produzidos neste módulo são passados ao próximo módulo (rede neural), para que sejam classificados.

4.2.3 Módulo Classificador

O módulo classificador tem como função classificar vetores de entrada, ou seja, *e-mails*, nas classes *ham* ou *spam*. Para isso, faz uso de um modelo neural artificial MLP treinado com *backpropagation*, descrito no capítulo 2.7.

4.3 Configurações XML

O SAS possui um arquivo XML de configuração que é lido antes da etapa de pré-processamento. A função deste arquivo é permitir a configuração do SAS. Até o presente momento, a configuração do arquivo XML permite apenas desabilitar os quatro métodos de pré-processamento com maior custo computacional — *vertical slicing, small table, invisible text* e *hidden words*. O formato XML foi escolhido devido às vantagens descritas na seção 2.4.

4.4 Base de dados

A base de dados é composta por *e-mails* reais, coletados no servidor de *e-mails* da Universidade Federal de Itajubá, em períodos de maior atividade. Todos os *e-mails*, à exceção do campo assunto (*subject*), tiveram os demais campos de seus cabeçalhos e assinaturas removidas, para garantir a privacidade dos mesmos.

A coleta foi realizada no dia 30 de julho de 2013, das 10 às 12 horas, no dia 31 de julho de 2013, das 14:38 às 16:38 horas, no dia 02 de agosto de 2013, das 11 às 13 horas, e no dia 12 de setembro, das 12 às 18 horas. Após a coleta e remontagem dos pacotes obteve-se 19.698 *e-mails*. Estes *e-mails* foram classificados manualmente, resultando em 4.988 *e-mails ham* e 14.710 *e-mails spam*.

O tráfego de pacotes de *e-mails* da porta de entrada do servidor da Universidade foi desviado de forma a possibilitar a coleta de aproximadamente 7.4 GB de dados referentes a *e-mails*. Posteriormente, utilizando o software Wireshark, os pacotes de dados coletados foram remontados em suas originais mensagens e, desta forma, os *e-mails* foram remonta-

dos em sua forma original. Na sequência, cada um dos 19.698 *e-mails* teve seu cabeçalho e assinatura removida e foi classificado manualmente em *ham* ou *spam*. Assim, pode-se aferir o percentual de classificação correta do sistema Barracuda e permitiu-se também que o SAS fosse testado com *e-mails* corretamente classificados.

O *software* Wireshark (WIRESHARK, 2013) foi escolhido por sua interface gráfica e por suas opções de filtragem de pacotes.

4.5 Rede Neural

A rede neural (MLP) foi configurada para testes com vetores de entrada de 1.000, 1.500, 2.000, 2.500 e 3.000 unidades de entrada. Após vários testes com diferentes combinações de neurônios nas camadas ocultas, a combinação de 10 neurônios na primeira camada oculta e 14 neurônios na segunda camada oculta foi a que apresentou melhores resultados. A camada de saída conteve unidades lineares para evitar *flat spots*, de acordo com Fahlman (FAHLMAN, 1988). Foi atribuída a função sigmoidal para ativação de cada perceptron da camada oculta. O treinamento foi feito baseado em épocas onde, ao final de cada época, a taxa de aprendizado e o momento foram modificados e o erro total calculado. O treinamento foi realizado com validação cruzada, isto é, ele é interrompido sempre que o erro total aumenta no conjunto de testes. O erro inicial empregado na fase de treinamento variou de 0,0001 a 0,00001. Para inicialização dos pesos, foi utilizado um valor aleatório entre [-0,5; 0,5]. Na saída da rede foram utilizadas duas unidades em todos os experimentos:

- (0 1) quando os padrões negativos são apresentados à entrada da rede, isto é, um *e-mail ham*.
- (1 0) quando os padrões positivos são apresentados à entrada da rede, isto é, um *e-mail spam*.

Para treinamento e teste da rede neural, a base de dados, contendo 19.698 *e-mails*, foi dividida em três conjuntos — treinamento, validação e teste. O conjunto de treinamento contém 40% dos *e-mails* da base, o conjunto de validação contém 20% dos *e-mails* da base e, por fim, conjunto de teste contém 40% dos *e-mails* da base.

4.6 Configuração dos Experimentos

Podem ocorrer casos nos quais a representação de um *e-mail* não possua nenhum dos *tokens* selecionados por algum dos três métodos de seleção de características. Neste caso, o vetor que representa o *e-mail* é um vetor nulo. Os padrões nulos foram removidos dos conjuntos de treinamento e validação. Foram, porém, mantidos no conjunto de teste.

O treinamento da rede neural por *backpropagation* exige que, nos conjuntos de treinamento, validação e teste, o número de padrões negativos (*ham*) seja igual ao de positivos (*spam*). Assim, antes da divisão dos *e-mails* em cada conjunto, os padrões *ham* foram duplicados aleatoriamente de forma a atingir o mesmo número de padrões *spam*.

Após a duplicação e divisão dos padrões, os conjuntos de testes e treinamento passaram a conter cada um 5.884 padrões negativos e 5.884 padrões positivos e o conjunto de validação passou a conter 2.942 padrões negativos e 2.942 padrões positivos.

Os experimentos realizados tiveram por objetivo verificar o desempenho do SAS com o uso de cada um dos três métodos de seleção de características — X^2 *Statistic*, DF e MI.

4.7 Experimentos

Foram realizados três experimentos, cada qual empregando um dos três métodos de seleção de características — X^2 *Statistic*, DF e MI. Em cada um dos três experimentos, a dimensionalidade do vetor de características variou de 1.000 a 3.000 características, com passos incrementais de 500 características. Para cada uma destas dimensionalidades do vetor de características, a rede neural foi testada por 10 vezes, onde cada teste utiliza os pesos finais obtidos em um treinamento. Os valores obtidos e listados nas tabelas dos experimentos representam a média dos 10 resultados obtidos e a variação representa o desvio entre o maior e o menor valor obtido nos 10 testes.

4.7.1 Experimento 1

O primeiro experimento empregou o método X^2 *Statistic*, descrito em 2.8.1, para seleção de características. A Tabela 1 apresenta o percentual de classificações corretas, em termos de *ham* e *spam*, obtido com cada dimensionalidade do vetor de características.

Tabela 1: *Desempenho percentual do SAS empregando* X^2 *Statistic*

Dim. Vetor	Ham	Spam
1.000	96,25 ± 0,55	96,35 ± 0,80
1.500	99,02 ± 0,74	91,88 ± 0,33
2.000	99,32 ± 0,62	86,82 ± 0,58
2.500	99,33 ± 0,49	89,74 ± 0,77
3.000	99,50 ± 0,45	99,09 ± 0,71

Conforme a dimensionalidade do vetor de características cresce, cresce, igualmente, o percentual de classificações corretas de *e-mails ham*.

Com o método X^2 *Statistic* e 3.000 características no vetor de características, a rede neural alcançou uma taxa de 99,50% de classificações corretas dos *e-mails* com conteúdo *ham*. Isto indica uma baixa taxa (0,50%) de falsos positivos.

A rede alcançou, igualmente, uma taxa de falsos negativos de 0,91%. Esta taxa situa-se abaixo de 5%, taxa definida como aceitável por Subramaniam *et al.* (SUBRAMANIAM *et al.*, 2010). A Figura 9 ilustra graficamente os resultados apresentados na Tabela 1.

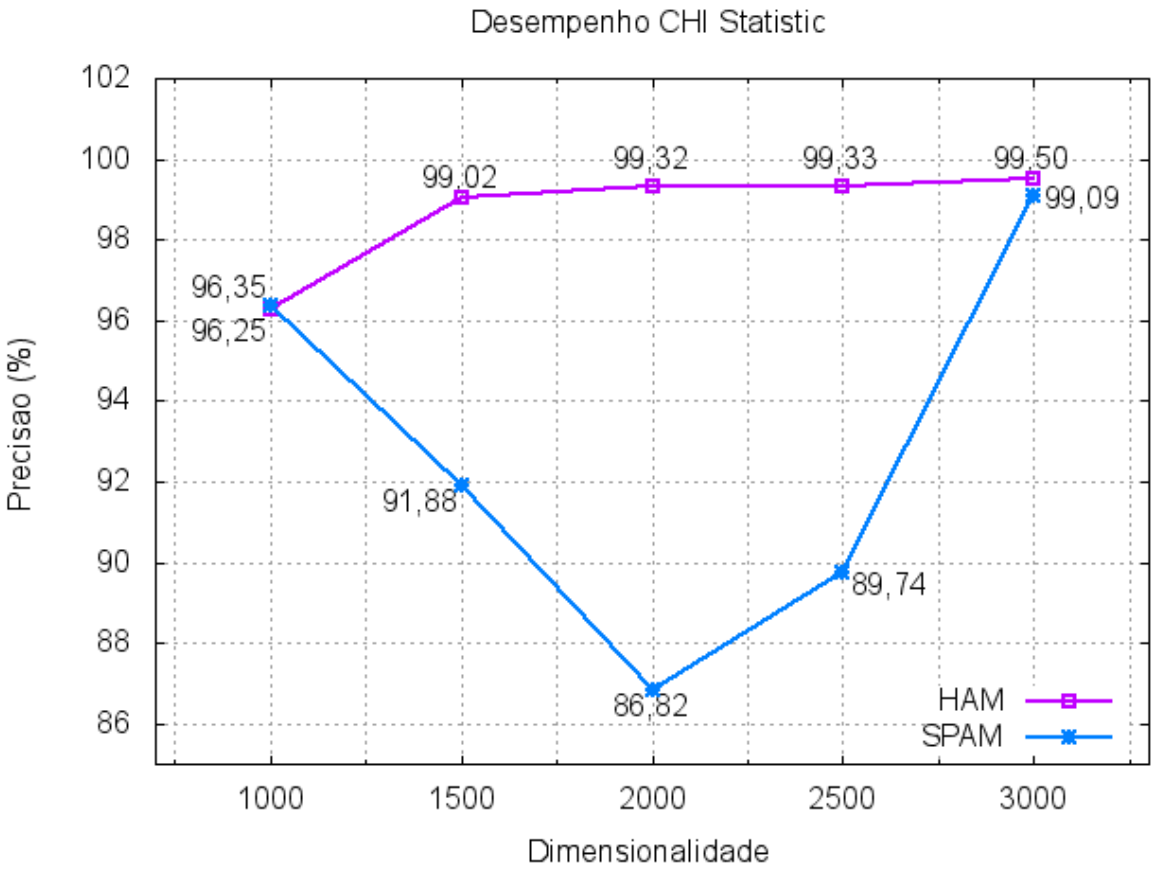

Figura 9: *Desempenho percentual do SAS empregando X^2 Statistic*

4.7.2 Experimento 2

O segundo experimento empregou o método *Frequency Distribution* (DF), descrito em 2.8.2, para seleção de características. A Tabela 2 apresenta o percentual de classificações corretas, em termos de *ham* e *spam*, obtido com cada dimensionalidade do vetor de características.

Tabela 2: *Desempenho percentual do SAS empregando DF*

Dim. Vetor	Ham	Spam
1.000	96,18 ± 0,54	96,37 ± 0,50
1.500	99,04 ± 0,94	92,12 ± 0,41
2.000	94,73 ± 0,65	99,01 ± 0,38
2.500	96,73 ± 0,63	98,99 ± 0,53
3.000	99,34 ± 0,55	99,11 ± 0,70

Com este método de seleção de características, a rede neural apresentou bons resultados, tanto na classificação de *e-mails hams* quanto de *spams*.

O aumento da dimensionalidade do vetor de características, na maioria dos casos, também ocasiona o aumento das taxas de classificação corretas. Tratando-se das taxas de classificação corretas de *e-mails spam*, a rede neural apresenta resultados superiores aos do experimento anterior.

A rede neural alcançou uma taxa de 99,34% de classificações corretas de *e-mails ham*, traduzindo-se em uma baixa taxa (0,66%) de falsos positivos. A rede alcançou, igualmente, uma taxa de falsos negativos de 0,89%, abaixo do valor 5%, definido como limite por Subramaniam *et al.* (SUBRAMANIAM *et al.*, 2010).

A Figura 10 ilustra graficamente os resultados apresentados na Tabela 2.

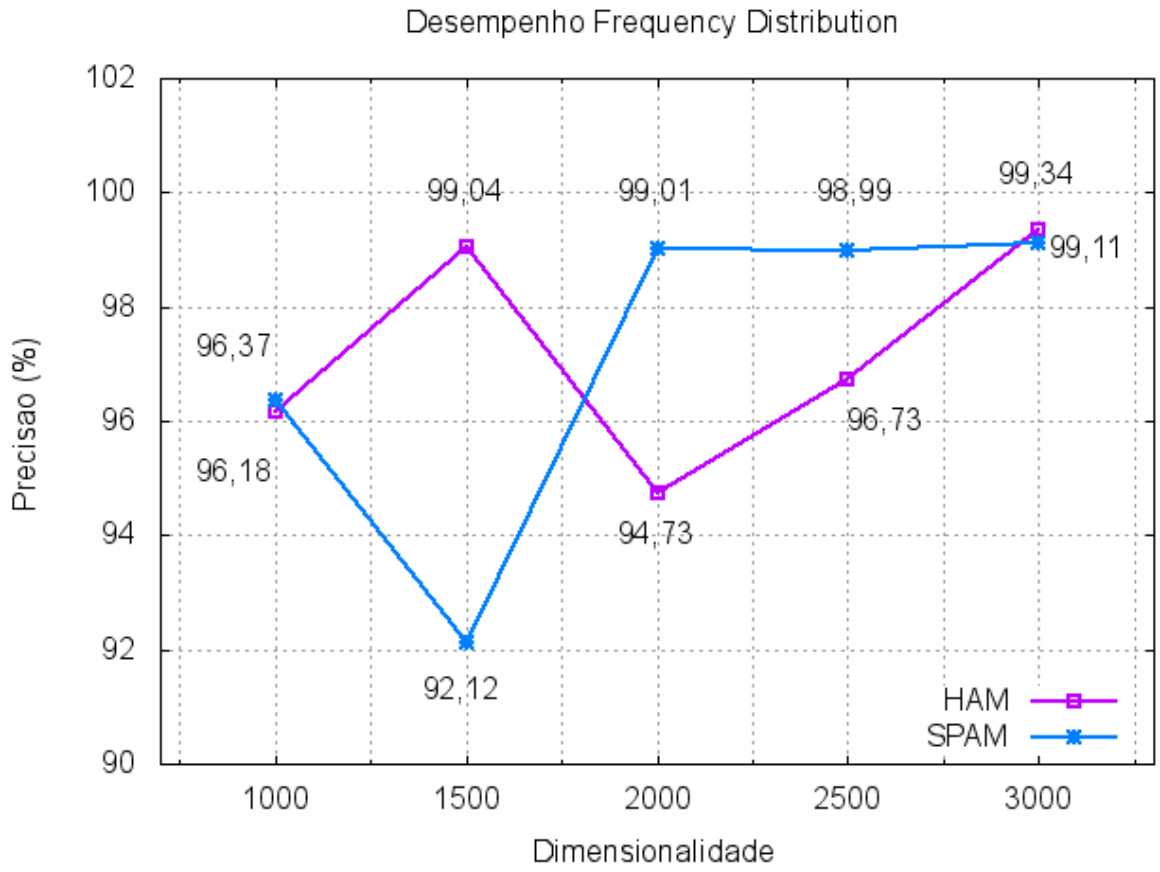

Figura 10: *Desempenho percentual do SAS empregando DF*

4.7.3 Experimento 3

O terceiro experimento empregou o método *Mutual Information* (MI), descrito em 2.8.3, para seleção de características. A Tabela 3 apresenta o percentual de classificações corretas, em termos de *ham* e *spam*, obtido com cada dimensionalidade do vetor de características.

Tabela 3: *Desempenho percentual do SAS empregando MI*

Dim. Vetor	Ham	Spam
1.000	96,52 ± 0,44	95,92 ± 0,78
1.500	99,21 ± 0,73	91,81 ± 0,32
2.000	99,05 ± 0,95	86,13 ± 0,41
2.500	99,15 ± 0,22	89,18 ± 0,23
3.000	99,31 ± 0,59	99,04 ± 0,84

Com o método MI, a rede neural apresentou bons resultados, semelhantes aos obtidos no primeiro experimento, principalmente quanto a estabilidade das taxas de classificação corretas de *e-mails ham* a partir de 1.500 características. Da mesma forma do primeiro experimento, o aumento da dimensionalidade do vetor de características também ocasiona o aumento das taxas de classificação corretas de *e-mails ham*.A rede alcançou taxas de 99,31% e 99,04% de classificações corretas de *e-mails ham* e *spam*, respectivamente. Assim, as taxas de falsos positivos e negativos foram de 0,69% e 0,96%, respectivamente. A Figura 11 ilustra graficamente os resultados apresentados na Tabela 3.

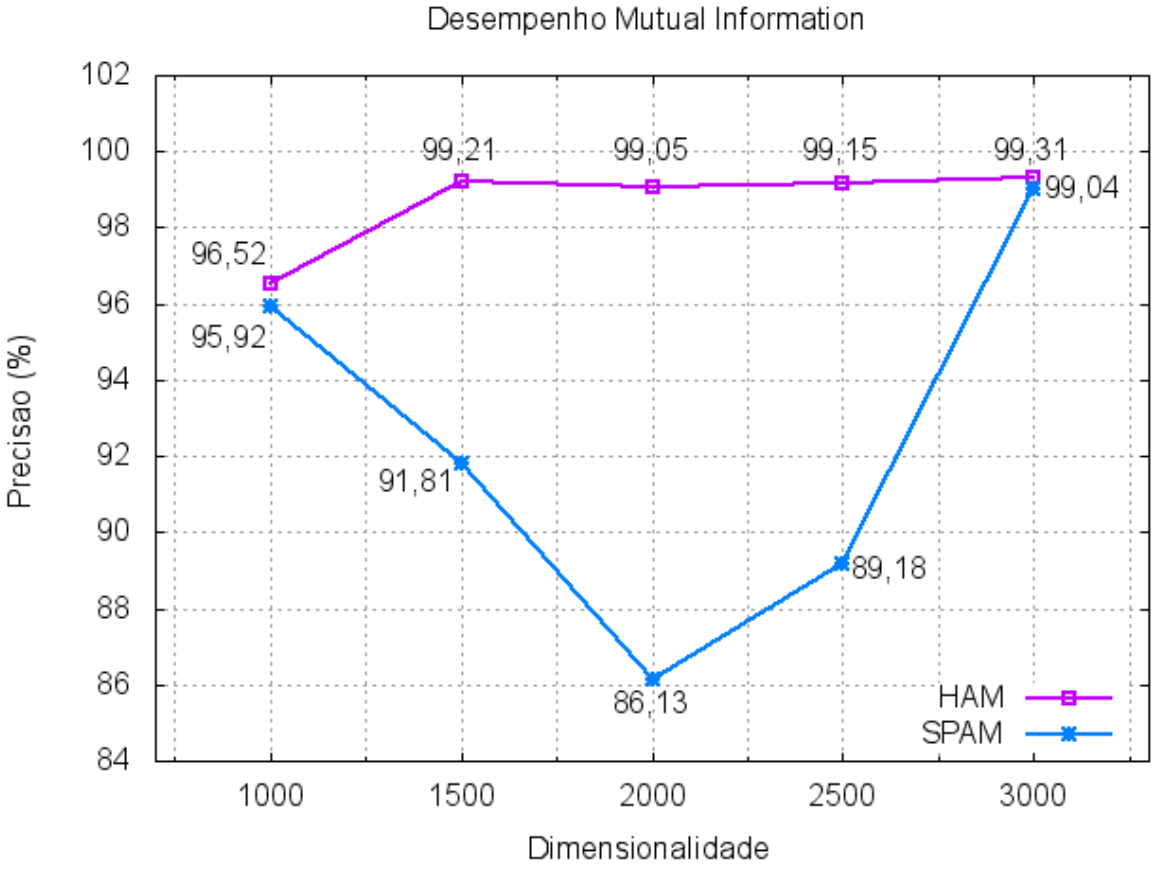

Figura 11: *Desempenho percentual do SAS empregando MI*

4.8 Análise dos Resultados

Os três métodos testados selecionam de forma eficiente as características mais relevantes dos *e-mails*. Com o uso destes métodos, a rede neural apresentou sempre bom desempenho tanto na classificação de *e-mails ham* quanto de *spam*. O gráfico da Figura 12 compara os resultados da rede neural, com o uso dos três métodos, na classificação de *e-mails ham*.

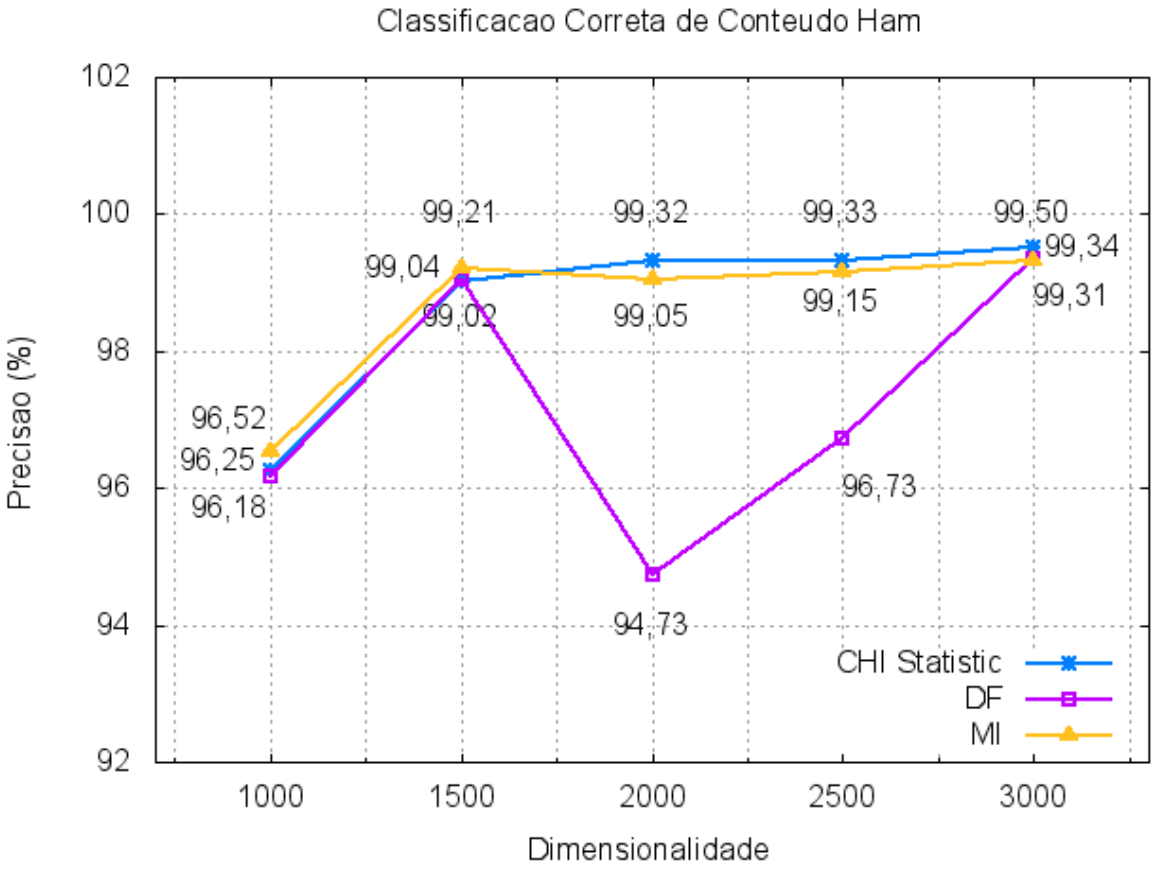

Figura 12: *Comparação dos resultados de classificação de hams*

Através da análise do gráfico, pode-se notar que, com os métodos X^2 *Statistic* e MI, a rede neural apresenta taxas de classificação correta de *hams* superiores a 96%. Por sua

vez, com o método DF, a rede somente não apresentou este mesmo desempenho com o vetor com dimensionalidade de 2.000 características.

As melhores taxas de classificação correta de *e-mails ham* foram de 99,50%, 99,34%, 99,31%, obtidas, respectivamente, com o uso dos métodos X^2 *Statistic*, DF e MI. Consequentemente, as taxas de falsos positivos foram de 0,50%, 0,66%, 0,69%, respectivamente. Com o uso dos três métodos, a rede neural apresentou, portanto, bons resultados.

Após a realização dos testes, o médoto MI foi o apresentou o melhor resultado na classificação correta de *e-mails ham* utilizando o menor número de características necessárias.

O gráfico da Figura 13 compara os resultados da rede neural, com o uso dos três métodos, na classificação de *e-mails spam*.

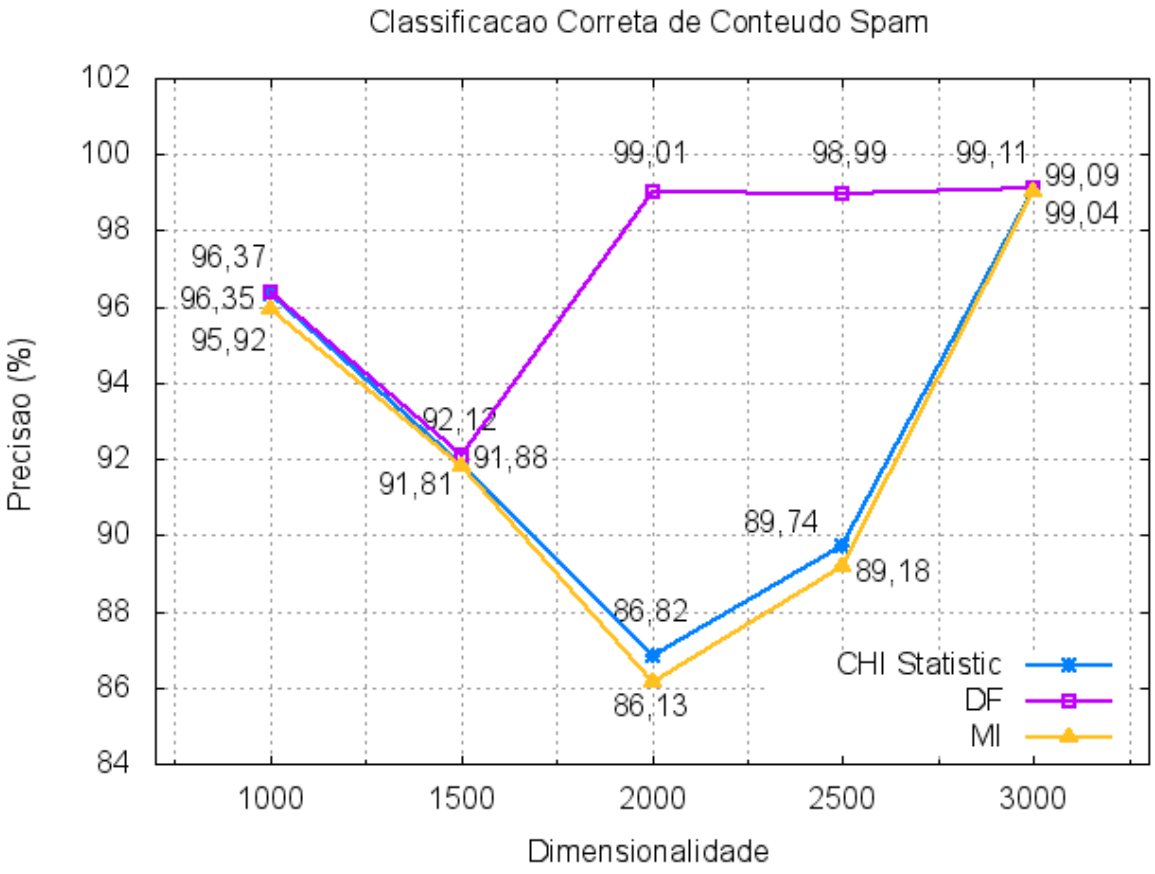

Figura 13: *Comparação dos resultados de classificação de spams*

Com o método DF, a rede neural apenas não apresenta taxa de classificação correta de *spams* superior a 96% com vetor com dimensionalidade de 1.500 características. Por sua vez, com o método X^2 *Statistic*, a rede apresenta taxas superiores a 96% apenas com vetores com dimensionalidades de 1.000 e 3.000 características. Com o método MI, porém, a rede passa a apresentar taxa superior a 96% somente com o vetor com dimensionalidade de 3.000 características. As melhores taxas de classificação correta de *e-mails spam* foram de 99,11%, 99,09%, 99,04%, obtidas, respectivamente, com o uso dos métodos DF, X^2 *Statistic* e MI. Consequentemente, as taxas de falsos negativos foram de 0,89%, 0,91%, 0,96%, respectivamente. Com o uso dos três métodos, a rede neural apresentou, igualmente, bons resultados.

Com o uso do método DF para seleção das características mais relevantes dos *e-mails*, a rede neural apresentou bons resultados tanto na classificação de *ham* quanto de *spam* com vetores de características de menor dimensionalidade. Este fato é relevante, pois reduz o custo computacional para treinamento e execução da rede neural, ou seja, do módulo classificador do SAS.

Após a realização dos testes, o médoto DF foi o apresentou o melhor resultado na classificação correta de *e-mails spam* utilizando o menor número de características necessárias.

Considerando as classificações corretas de *ham* e *spam*, o método X^2 *Statistic* pode ser o considerado o melhor método de seleção de características, pois obteve os melhores resultados classificatórios com 3.000 características.

Com vetores com dimensionalidade de 3.000 características, as taxas obtidas pela rede neural passam a ser similares, independentemente do método de seleção de características empregado. Além disto, as taxas passam, igualmente, a ter um crescimento insignificante. Assim, 3.000 características são suficientes para representar satisfatoriamente todos os *e-mails* da base utilizada nos experimentos.

Observa-se que em ambas as classificações os métodos apresentaram, em um determinado momento, uma queda no desempenho da classificação conforme o número de características do vetor de entradas da rede neural aumentou. Os métodos X^2 *Statistic* e MI apresentaram uma queda na classificação correta de *e-mails spam* quando a dimensio-

nalidade do vetor de características foi de 1.500 e 2.000 e somente voltou a subir com 2.500 características. O método DF obteve uma queda na classificação correta de *e-mails spam* com 1.500 características e uma queda na classificação correta de *e-mails ham* quando a dimensionalidade do vetor foi de 2.000 características.

Estas quedas, porém, são indiretamente proporcionais. Ao serem analisadas as classificações corretas de *e-mails ham* e *e-mails spam* em um determinado método, quando o desempenho na classificação correta de *e-mails spam* diminui, o desempenho na classificação correta de *e-mails ham* aumenta. O contrário também ocorre, quando o método DF apresenta a queda na classificação correta de *e-mails ham*, o desempenho na classificação correta de *e-mails spam* aumenta.

Por se tratarem de métodos estatísticos de seleção de características, tais métodos precisam selecionar os termos que refletem um real benefício de classificação. Isto indica que com determinadas dimensionalidades do vetor de características, os métodos selecionaram mais características relevantes a classificação de *e-mails spam* ou *e-mails ham*. Como os métodos X^2 *Statistic* e MI apresentam muita similaridade na escolha de características, apresentaram também um comportamento similar de desempenho, possuindo inclusive as mesmas quedas, pois selecionaram mais características relevantes a classificação de *e-mails ham*. Como o método DF seleciona apenas a frequência dos termos relevantes, este método apresentou um comportamento um pouco diferente, o que resultou no aumento do desempenho na classificação de *e-mails spam* e uma queda na classificação de *e-mails ham*.

Quando a dimensionalidade do vetor de entradas da rede neural foi de 3.000 características, os métodos de seleção de características foram capazes de selecionar as características mais relevantes tanto para a classificação correta de *e-mails ham* quanto para *e-mails spam*. É importante perceber que com um número reduzido de características — 1.000 características — todos os métodos estudados apresentaram desempenho acima de 90%.

Na análise de aproximadamente 10% dos *e-mails* classificados como falsos positivos, ou seja, dos *e-mails ham* classificados erroneamente como *spam* pelo SAS, verificou-se que, embora *hams*, apresentavam características *spam* muito marcantes, tais como, respostas a usuários contendo informações ou anúncios de produtos vendidos em *sites* de comércio

eletrônico, mensagens com diversos erros de digitação, mensagens de usuários reencaminhando a outros usuários *e-mails* promocionais de um *site* de compras, dentre outras similares. Assim, a classificação errônea deste *e-mails* pode ser razoavelmente justificada.

4.9 Resultados do Barracuda

O sistema anti-*spam* Barracuda classificou os mesmos *e-mails* da base empregada nos três experimentos acima descritos. Obteve taxas de 99,49% e 31,89% na classificação correta de *hams* e de *spams*, respectivamente.

O filtro anti-*spam* Barracuda apresenta, assim, uma baixa taxa de falsos positivos (0,51%). A taxa de falsos negativos é, porém, de 68,11%, um valor muito pobre, bem pior que o considerado satisfatório (inferior a 5%) por Subramaniam *et al.* (SUBRAMANIAM *et al.*, 2010).

4.10 Comparação entre SAS e Barracuda

Como apresentado anteriormente na seção 4.9, o Barracuda obteve uma boa taxa de classificação correta de *e-mails ham*, produzindo, assim, uma taxa de apenas 0,51% de falsos positivos.

Os melhores resultados de falsos positivos obtidos pelo SAS foram de 0,50%, 0,66% e 0,69%, com o uso dos métodos X^2 *Statistic*, DF e MI, respectivamente. Assim, em termos de taxas de falsos positivos e de classificações corretas de *e-mails ham*, o SAS e o Barracuda apresentaram desempenho similar.

Em termos de classificação correta de *e-mails spam*, porém, o Barracuda apresentou um resultado ruim, pois sua taxa de falsos negativos foi de 68,11%.

Os melhores resultados de falsos negativos obtidos pelo SAS foram de 0,89%, 0,91% e 0,96%, com o uso dos métodos X^2 *Statistic*, DF e MI, respectivamente. Assim, em termos de taxas de falsos negativos e de classificações corretas de *e-mails spam*, o SAS apresentou

um desempenho muito superior ao do Barracuda.

O desempenho inferior do Barracuda, em termos de taxa de falsos negativos, pode ser devido a diversos fatores. Como possui código proprietário e fechado, porém, não é possível determinar qual ou quais fatores foram determinantes para este baixo desempenho.

Por fim, independentemente dos resultados alcançados pelo Barracuda, é necessário enfatizar que o SAS, fazendo uso de seus três módulos — Pré-Processamento, Seleção de Características e Classificador —, é capaz de alcançar bons resultados em um ambiente real.

5 Conclusão

5.1 Considerações finais

Este trabalho de dissertação propõe um sistema anti-*spam* (SAS) de três módulos — Pré-Processamento, Seleção de Características e Classificador — para a classificação de *e-mails* como *ham* ou *spam*.

O módulo de Pré-Processamento analisa o conteúdo dos *e-mails*, identificando palavras, *tags* HTML e padrões de ofuscamento, para produzir unidades de informação, denominadas *tokens*. Os *tokens* são passados ao módulo de seleção de características.

O módulo de Seleção de Características é responsável por selecionar os *tokens* mais relevantes para a classificação dos *e-mails*. Três métodos estatísticos de seleção de características — X^2 *Statistic*, MI e DF — foram empregados neste trabalho de dissertação. O uso de métodos de seleção de características reduz a dimensionalidade dos vetores de características que representam os *e-mails*, reduzindo, assim, o custo computacional para treinamento e execução do módulo classificador neural.

O módulo Classificador é composto por uma rede neural MLP. Recebe, do módulo anterior, cada *e-mail*, representado por seu vetor de características, e o classifica como *ham* ou *spam*.

O SAS foi avaliado sobre uma base contendo *e-mails* reais, coletados durante períodos distintos na Universidade Federal de Itajubá. Seus resultados foram comparados aos produzidos, sobre esta mesma base, pelo sistema anti-*spam* comercial Barracuda, utilizado na universidade de 2005 a 2013.

A base de *e-mails* contém 19.698 *e-mails*, dos quais 4.988 são *hams* e 14.710 são *spams*. Os *e-mails* possuem características diversas, tais como anexos, idiomas e codificações diferentes, representando o contexto usual das mensagens recebidas pelos usuários da universidade.

O Barracuda produziu taxas de classificação correta de *e-mails spam* e *ham* de 31,89% e 99,49%, respectivamente. Estes valores foram obtidos através da análise manual de cada *e-mail* classificado.

Os resultados alcançados pelo SAS foram bons, acima de 90%. Com o uso do método X^2 *Statistic*, a rede neural alcançou a taxa de 99,50% na classificação correta de *e-mails* com conteúdo *ham*, o que representa uma taxa de falsos positivos de 0,50%. Em termos de classificação de *e-mails* com conteúdo *spam*, a rede alcançou uma taxa de 99,09%, o que se traduz em uma taxa de falsos negativos de 0,91%.

Com o uso do método DF, a rede neural alcançou taxas de 99,34% e 99,11% na classificação correta de *e-mails* com conteúdo *ham* e *spam*, respectivamente. Estes valores traduzem-se em taxas de 0,66% e 0,89% de falsos positivos e negativos, respectivamente.

Com o uso do método MI, a rede neural alcançou taxas de 99,31% e 99,04% na classificação correta de *e-mails ham* e *spam*, respectivamente. Assim, as taxas de falsos positivs e negativos foram, respectivamente, de 0,69% e 0,96%.

Desta forma, o método X^2 *Statistic* foi o que apresentou os melhores resultados em relação aos três métodos escolhidos com a maior dimensionalidade do vetor, ou seja, 3.000 características. O método DF apresentou os melhores resultados com menos características — 1.000 características. Estas informações são relevantes porque o número de características está diretamente relacionado ao desempenho do sistema, assunto não abordado por esta dissertação.

Em termos de taxas de classificação correta de *e-mails spam* e de falsos positivos, os resultados alcançados pelo SAS são similares aos produzidos pelo Barracuda. Em termos de taxas de classificação correta de *e-mails ham* e de falsos negativos, porém, o desempenho do Barracuda situou-se bem abaixo do alcançado pelo SAS, com o uso de qualquer um dos 3 métodos de seleção de características.

Assim, o objetivo deste trabalho foi atingido, uma vez que todo trabalho proposto na Seção 1.6 foi realizado. O novo módulo de Pré-Processamento do SAS foi atualizado e agora é capaz de lidar com uma gama complexa de novas técnicas de ofuscamento de *e-mails*, o módulo Classificador foi desenvolvido com sucesso na linguagem Java e, com isto, todos os três módulos foram integrados tornando o SAS um sistema único. As configurações do SAS agora podem ser habilitadas ou desabilitadas através de um arquivo XML e os resultados obtidos nesta nova versão do SAS foram comparados ao do sistema comercial Barracuda.

Por fim, independentemente dos resultados alcançados pelo Barracuda, é necessário enfatizar que o SAS, fazendo uso de seus três módulos — Pré-Processamento, Seleção de Características e Classificador —, é capaz de alcançar bons resultados em um ambiente real.

5.2 Trabalhos Futuros

Como sugestões para desenvolvimentos futuros deste trabalho, podem-se mencionar as seguintes:

- *Análise de anexos*: Arquivos anexados a *e-mails* podem conter *softwares* maliciosos (vírus, *trojans* e semelhantes). Estes anexos poderiam, no futuro, ter seus conteúdos analisados por um anti-vírus, o que evitaria a proliferação destes *softwares* maliciosos;

- *Análise de imagens*: Atualmente, *spammers* fazem uso de imagens com conteúdo *spam* em *e-mails*. Como imagens não possuem codificação textual, o SAS bem como os demais sistemas anti-*spam* existentes, não conseguem analisar e classificar as imagens. Assim, no futuro, seria recomendável a implementação de um novo módulo no SAS, para análise de imagens;

- *Algoritmos de seleção negativa*: Tal como em um sistema imunológico artificial (MA *et al.*, 2009), a rede neural MLP implementada no módulo classificador pode ser substituída por um classificador de uma única classe, no caso, de *e-mails ham*. O uso deste classificador pode trazer duas vantagens. Primeira, posto que o número de *e-mails ham* circulando na *Internet* é bem inferior ao de *e-mails spam*, o universo

de *e-mails* a ser classificado é menor e, por conseguinte, menores serão também os custos computacionais para treinamento e execução do classificador. A segunda vantagem consiste no fato do classificador poder, em teoria, ser capaz de adaptar-se automaticamente às novas técnicas utilizadas por *spammers*, sem haver necessidade de incorporá-las aos conjuntos de treinamento do classificador;

- *Novos testes em ambiente real*: Atualmente, a Universidade Federal de Itajubá substituiu seu sistema anti-*spam* Barracuda pelo sistema também comercial Canit Anti*Spam*. Uma nova comparação dos resultados produzidos pelo SAS e pelo Canit pode ser, portanto, realizada;

- *Novos métodos de seleção de características*: Novos métodos estatísticos de seleção de características podem ser implementados no segundo módulo do SAS, de forma a verificar-se se aumentam o desempenho do módulo classificador ou reduzem seu custo computacional;

- *Módulo de treinamento contínuo*: Seria interessante, no futuro, implementar no SAS um módulo de treinamento contínuo, supervisionado pelos usuários. Com isto, os usuários seriam capazes de corrigir classificações incorretas do SAS, de forma a melhorar seu desempenho nas classificações;

- *Incorporação de técnicas estáticas*: O anti-*spam* Barracuda faz uso de técnicas estáticas (listas brancas e negras) para classificação dos *e-mails*. O servidor de *e-mails* de código livre *Exim Internet Mailer*[1] implementa o acesso a informações constantes em listas negras disponíveis na *Internet*. Seria interessante, portanto, tornar o Exim um módulo do SAS, de forma a que este possa, além de técnicas dinâmicas, também contar com técnicas estáticas para classificação de *e-mails*;

- *Licença e distribuição*: Este trabalho de pesquisa tem, por objetivo final, o desenvolvimento de um produto tecnológico sob licença livre.

[1] http://www.exim.org

Referências

ANDROUTSOPOULOS I., GEORGIOS G., PALIOURAS G., KARKALETSIS V., SPYROPOULOS C. D., and STAMATOPOULOS P. *A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists.* Information Retrieval, vol. 6, pp. 49-73, 2003.

ANTISPAM. *Técnicas de Spammers.* Disponível em: <http://www.antispam.br/>. Acessado em 11 de julho de 2013.

BACHMANN D. and ELFRINK J. *Tracking the progress of e-mail versus snail mail.* Marketing Research, ed.1, pp. 31-35, 1996.

BISHOP C. M. *Neural Netword for Patern Recognition.* Clarendon Press, Oxford, 1995.

BISHOP C. M. *Exact Calculation of the Hessian Matrix for the Multilayer Perceptron. Neural Computation.* Journal Neural Computation, vol. 4, pp. 494-501, 1992.

BRAGA A. P., CARVALHO A. C. P. L. F., and LUDEMIR T. B. *Redes Neurais Artificiais , Teoria e Aplicações.* TLC Editora, ed. 1, 2007.

Bravo Tecnologia. *Barracuda Sistema Anti-Spam.* Disponível em: <http://www.bravotecnologia.com.br/Barracuda/>. Acessado em 11 de julho de 2013.

CAPANEMA W. A. *Spam e as Pragas Digitais.* LTR Editora, ed. 1, 2009.

CHUAN Z., XIANLIANG L., MENGSHU H., and XU Z. *A LVQ-based neural network anti-spam e-mail approach.* ACM SIGOPS Operating Systems Review, vol. 39, pp. 34-39, 2005.

CLARK J., KOPRINSKA I., and POON J. *A Neural Network Based Approach to Automated E-mail Classification.* IEEE/WIC International Conference on Web Intelligence, pp. 702-705, 2003.

COURNANE A. and HUNT R. *An analysis of the tools used for the generation and prevention of spam.* Computers and Security, vol. 23, Issue 2, pp. 154-166, 2004.

CRIVISQUI E. *Apresentação de métodos de classificação.* Seminário de métodos estatísticos mutivariados aplicados as ciências humanas, pp. 1-57. Unicamp, Campinas, Brasil, 1998.

GOETSCHI R. *Spam Filtering Using Artificial Networks.* Semester Thesis, Computer Science Department. Berne University of Applied Sciences, Bern, Suíça, 2004.

HAYKIN S. *Redes Neurais Princípios e Práticas*. Bookman Editora, ed. 2, 2000.

FAHLMAN S. E. *An empirical study of learning speed in backpropagation networks*. Technical Report CMU CS 88 162. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1988.

GOWEDER M., RASHED T., ELBEKAIE A. S., and HUSIEN A. A. *An Anti-Spam System Using Artificial Neural Networks and Genetic Algorithms*. Proceedings of the 2008 International Arab Conference on Information Technology, pp. 1-8, 2008.

JOACHIMS T. *Text categorization with support vector machines: learning with many relevant features in Machine Learning*. ECML-98, 10th European Conference on Machine Learning, vol. 1398, pp 137-142, 1998.

KUSHMERICK N. and THOMAS B. *Adaptive information extraction: Core technologies for information agents. Publicado no livro Intelligent Information Agents.* Springer Editora, ed. 1, pp 79-103, 2003.

LI S.M. and CHUNG T.M. *Internet function and Internet addictive behavior*. Computers in Human Behavior, vol. 22, Issue 6, pp. 1067-1071, 2006.

MA W., TRAN D., and SHARMA D. *A Novel Spam Email Detection System Based on Negative Selection*. 4a. International Conference - Computer Sciences and Convergence Information Technology (ICCIT '09), pp. 987-992, 2009.

MCDONALD L. *Transactional E-mails: Make Your First Impression Count*. Disponível em: <http://www.mediapost.com/publications/article/104687/#axzz2dTLv00eB>. Acessado em 11 de julho de 2013.

MCKINSEY. *The social economy: Unlocking value and productivity through social technologies*. Disponível em: <http://www.mckinsey.com/insights/high_tech_telecoms_Internet/the_social_economy>. Acessado em 10 de julho de 2013.

MUELLER S.H. *What is spam?* Disponível em: <http://spam.abuse.net/overview/whatisspam.shtml>. Acessado em 12 de julho de 2013.

NPR. *At 30, spam Going Nowhere Soon*. Entrevista realizada por Gary Thuerk e Joel Furr, 2008.

PARK S. and AN D. U. *Automatic E-mail Classification Using Dynamic Category Hierarchy and Semantic Features*. IETE Technical Review, vol. 27, Issue 6, pp. 478-492, 2010.

PARK S., PARK S. H., LEE J. HONG., and LEE J. S. *E-mail Classification Agent Using Category Generation and Dynamic Category Hierarchy*. Artificial Intelligence and Simulation, vol. 3397, pp. 207-214, 2005.

PARTRIDGE C. *The Technical Development of Internet E-mail*. Annals of the History of Computing, vol. 30, Issue 2, pp. 3-29, 2008.

ROYAL PINGDOM. *Internet 2012 in numbers*. Disponível em: <http://royal.pingdom.com/2013/01/16/internet,2012,in,numbers/>. Acessado em 12 de julho de 2013.

SABRI A. T., MOHAMMADS A. H., AL-SHARGABI B., and HAMDEH M. A. *Developing New Continuous Learning Approach for Spam Detection using Artificial Neural Network*. European Journal of Scientific Research, vol. 42, Issue 3, pp. 511-520, 2010.

SEBASTIANI F. *Machine learning in automated text categorization*. ACM Computing Surveys (CSUR), vol. 34, Issue 1, pp. 1-47, 2002.

SHI L., WANG Q., MA X., WENG M., and QIAO H. *Spam Email Classication Using Decision Tree Ensemble*. Journal of Computational Information Systems, vol. 8, pp. 949-956, 2012.

SILVA A. M., MOITA G. F., and ALMEIDA P. E. M. *Detecção de SPAM utilizando Redes Neurais Artificiais SOM*. Master Thesis, CEFET-MG. Laboratório de Sistemas Inteligentes, Contagem, Minas Gerais, Brasil, 2010.

STUART I., Cha S. H., and TAPPERT C. *A Neural Network Classifier for Junk E-Mail*. Document Analysis Systems VI - 6th International Workshop, vol. 3163, pp. 442-450, 2004.

SUBRAMANIAM T., JALAD H. A., and TAQA A. Y. *Overview of textual anti-spam filtering techniques*. International Journal of the Physical Sciences, vol. 5, pp. 1869-1882, 2010.

TAK G. K. and TAPASWI S. *Query Based Approach Towards Spam Attacks Using Artificial Neural Network*. International Journal of Artificial Intelligence and Applications, vol. 1, Issue 4, pp. 82-95, 2010.

TAKASHITA T., TSUYOSHI I., TERUAKI K., and MASAYOSHI A. *Extracting user preference from Web browsing behaviour for spam filtering Int. J. Advanced Intelligence Paradigms*. International Journal of Advanced Intelligence Paradigms, vol. 1, pp. 126-138, 2008.

TEIXEIRA R. C. *Combatendo o Spam*. Novatec Editora, ed. 1, 2004.

VAN VLECK T. *The History of Electronic Mail*. Disponível em: <http://www.multicians.org/thvv/mail-history.html>. Acessado em 11 de julho de 2013.

W3C. *About XML*. Disponível em: <http://www.w3.org/XML/>. Acessado em 11 de julho de 2013.

WIRESHARK. *Sobre o Wireshark*. Disponível em: <http://www.wireshark.org/about.html>. Acessado em 11 de julho de 2013.