RICARDO DE PAULA POUÇAS

MODELOS EVOLUTIVOS BASEADOS EM GRÂNULOS E NUVENS DE DADOS PARA CLASSIFICAÇÃO ONLINE DE SPAM

LAVRAS – MG

2017

RICARDO DE PAULA POUÇAS

MODELOS EVOLUTIVOS BASEADOS EM GRÂNULOS E NUVENS DE DADOS PARA CLASSIFICAÇÃO ONLINE DE SPAM

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, área de concentração em Inteligência Computacional, para a obtenção do título de Mestre.

Prof. DSc. Daniel Furtado Leite

Orientador

LAVRAS – MG

2017

Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Pouças, Ricardo de Paula.
Modelos Evolutivos baseados em Grânulos e Nuvens de Dados para Classificação Online de Spam / Ricardo de Paula Pouças. - 2017.
101 p.

Orientador(a): Daniel Furtado Leite.

Dissertação (mestrado acadêmico) - Universidade Federal de Lavras, 2017.
Bibliografia.

1. Detecção de Spam. 2. Sistemas Inteligentes Evolutivos. 3. Sistemas Fuzzy. I. Leite, Daniel Furtado. II. Título.

RICARDO DE PAULA POUÇAS

MODELOS EVOLUTIVOS BASEADOS EM GRÂNULOS E NUVENS DE DADOS PARA CLASSIFICAÇÃO ONLINE DE SPAM

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação, área de concentração em Inteligência Computacional, para a obtenção do título de Mestre.

APROVADA em 25 de agosto de 2017.

Prof. DSc. Maury Meirelles Gouvêa Junior PUC–MG/PPGEE
Prof. DSc. Demóstenes Zegarra Rodríguez UFLA/DCC

Prof. DSc. Daniel Furtado Leite
Orientador

LAVRAS – MG
2017

Dedico este trabalho a minha família, e todos aqueles que contribuíram direta ou indiretamente para que eu atravessasse esta jornada.

AGRADECIMENTOS

À Universidade Federal de Lavras (UFLA), ao Departamento de Engenharia (DEG) e ao Programa de Pós-Graduação em Engenharia de Sistemas e Automação (PPGESISA), pela oportunidade concedida.
Ao Prof. Daniel Furtado Leite, pelo apoio e orientação.
Aos professores do PPGESISA , pela contribuição e embasamento teórico.
Aos amigos do PPGESISA, que sempre me ajudaram a superar as dificuldades e não desistir.
A todos os professores que contribuíram para minha formação acadêmica.
Aos mestres supremos que me propiciaram esta grande experiência.

RESUMO

Enviar e receber e-mails tem se tornado um problema devido ao fato de que pessoas mal-intencionadas utilizam essa ferramenta para disseminar códigos maliciosos com o objetivo de infectar computadores ou roubar informação. O ato de enviar e-mails sem a permissão do usuário é denominado spam. Existem várias técnicas para disseminação de spam. Elas são baseadas no conteúdo da mensagem ou em alguma fragilidade do sistema classificador que tenta interceptar mensagens. Sistemas classificadores capazes de se auto adaptar continuamente conforme a necessidade são raros. A necessidade de adaptação se dá visto às características variáveis de spams como consequência do uso de diversas técnicas de mascaramento de mensagem. Além disso, modelos classificadores que lidam com grandes volumes de dados utilizando o menor custo computacional possível são interessantes. Sistemas Inteligentes Evolutivos são capazes de se adaptar parametricamente e estruturalmente frente às mudanças em um fluxo de dados extraído de e-mails. Neste trabalho foi utilizado o método TEDA (*Typicality and Eccentricity based Data Analytics*) e o método FBeM (*Fuzzy Set-Based Evolving Modeling*) para classificação de spam online de forma não supervisionada. TEDA é um método que se baseia nos conceitos de nuvem de dados, excentricidade e tipicidade. A ideia é que nuvens TEDA não têm um formato geométrico específico, como clusters convencionais. FBeM usa objetos fuzzy granulares para sumarizar a informação extraída de um fluxo. FBeM é baseado no conceito de cobertura (granulação) do espaço dos dados. Suas regras são interpretáveis linguisticamente; elas são úteis para auxílio à tomada de decisão. Os métodos TEDA e FBEM são comparados em termos do erro de classificação, custo computacional e parcimônia. Para redução de dimensionalidade foi utilizado o algoritmo ACO (*Ant Colony Optimization*). ACO se trata de um algoritmo inspirado na inteligência do comportamento de formigas. O problema de seleção de variáveis é representado em um grafo, onde um caminho ótimo minimiza uma função objetivo e sugere variáveis mais discriminativas de e-mails spam. Uma base de dados contendo 25745 amostras, sendo 7830 spams e 17915 e-mails legítimos, foi criada. Cada amostra é descrita por 711 variáveis extraídas de um servidor de e-mails.

Palavras-chave: Detecção de Spam. Sistemas Inteligentes Evolutivos. Sistemas Fuzzy. Agrupamento Incremental. Nuvem de Dados.

ABSTRACT

Sending and receiving e-mails has become a concern since people use such tool to disseminate malicious code aiming to damage a computer system or steal information. The act of sending a message without user permission is called spam. There exist several techniques to disseminate spams. They are based on the content of the message or in some weakness of the classification system, which intercepts messages. Classification systems able to self-adapt over time are rare. Adaptation is needed because spams vary over time as consequence of the application of several message-masking techniques. Moreover, classification models that handle large volumes of data using low computational resource are interesting. Evolving Intelligent Systems are able to adapt their parameters and structure in view of the changes in a stream of data extracted from e-mails. This work uses TEDA (Typicality and Eccentricity based Data Analytics) and FBeM (Fuzzy Set-Based Evolving Modeling) for online unsupervised classification of spams. TEDA is based on the concepts of data clouds, eccentricity and typicality. The idea is that TEDA clouds do not have a specific geometric shape such as conventional clusters. FBeM uses fuzzy granular objects to summarize information extracted from a data stream. FBeM is based on the concept of coverage (granulation) of the data space. Its rules are linguistically interpretable; they are useful to help decision making. TEDA and FBeM are compared in the sense of classification error, processing speed and parsimony. For dimensionality reduction, ACO (Ant Colony Optimization) is employed. ACO is inspired on intelligent behavior of ants. The feature selection problem is represented as a graph, where the optimum path minimizes an objective function and suggests the most discriminate features for spam classification. A dataset containing 25745 samples, being 7830 spams and 17915 legitimate e-mails, was created. 711 features extracted from an e-mail server describe each sample.

Keywords: Spam Detection. Evolving Intelligent Methods. Fuzzy systems.
Incremental Clustering. Data Clouds.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O *email* é um dos serviços mais antigos da Internet e também é um dos principais meios para envio de mensagens não solicitadas, conhecidas como spam. Devido a grande quantidade de técnicas de envio de mensagens, que variam de acordo com a cultura, época, entre outros fatores externos, o problema de detecção de spam ainda está longe de ser solucionado por completo. É importante que sejam realizadas pesquisas a fim de encontrar uma solução capaz de permitir a correta classificação de *emails* originais e spams.

O protocolo SMTP (*Simple Mail Transfer Protocol*) é o mais utilizado para envio de mensagens eletrônicas (KLENSIN, 2008). Por ser um dos serviços mais utilizados na Internet, quando o assunto é envio de mensagens, o *email* se destaca por ser um serviço gratuito do ponto de vista do usuário, e barato do ponto de vista das empresas. Por este motivo, o *email* se tornou um dos principais meios de envio de mensagens de *marketing*, e paralelamente, tornou-se também, um dos principais meios de envio de mensagens fraudulentas. O objetivo destas mensagens é obter alguma vantagem ilícita ou enviar códigos maliciosos.

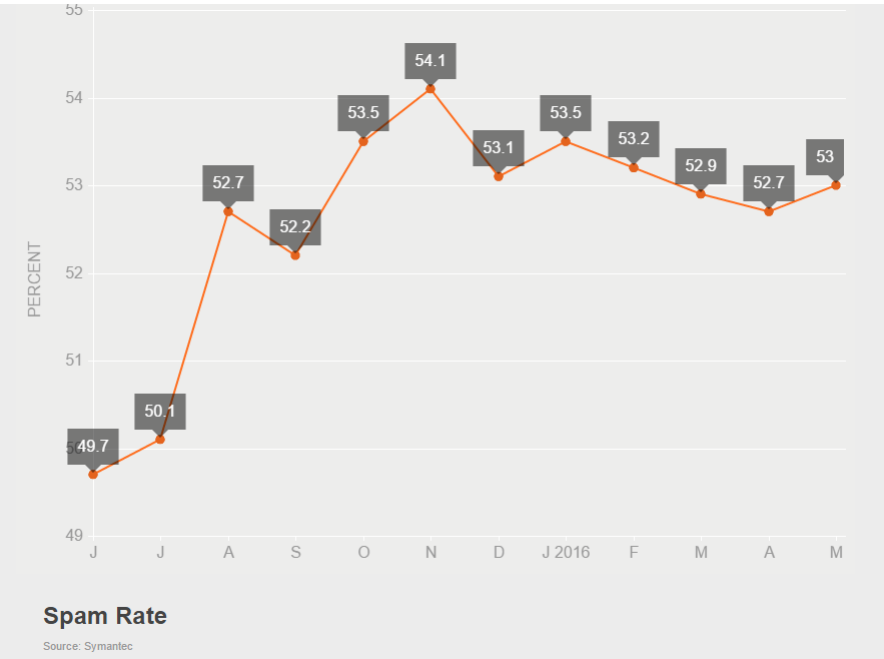
Envio de spam através de *emails* tem causado vários problemas que vão desde o aborrecimento de usuários, que encontram dificuldade devido ao acúmulo de mensagens irrelevantes em suas caixas de entrada, até ao ponto de sobrecarregar um servidor SMTP, devido ao grande volume de mensagens. Em muitos casos, o dano excede gravemente os motivos citados acima chegando ao ponto de comprometer os dados dos usuários ou empresas e gerar prejuízos morais e financeiros (JAKOBSSON, 2005).

O primeiro envio de spam ocorreu em 1978, quando foi enviado uma mensagem para 393 usuários da primeira rede experimental chamada ARPANET (OLIVO; SANTIN; OLIVEIRA, 2010). Dessa época em diante, os dados estatísticos sobre o envio de spam mostraram-se cada vez mais preocupantes. A *AOL* (América Online) avaliou que cerca de 5 a 30% de seus *emails*, recebidos diariamente, de um volume de cerca de 10 milhões, eram spam (HOANCA, 2006). De acordo com (WHITWORTH; WHITWORTH, 2004), em uma escala global, a porcentagem desse tipo de mensagem chega a 95% do total de mensagens trafegadas.

Um estudo estatístico divulgado pela Symantec revelou que em 2010 o percentual de spams chegou a 89,1%do número total de *emails* enviados (FOSSI et al., 2011), sendo este o maior percentual já atingido (FOSSI et al., 2011). Em 2010, a quantidade de mensagens eletrônicas do tipo spam foi de cerca de 62 bilhões. Entretanto, a quantidade de spams no ano 2013 recuou para 60% do número total de mensagens, o que corresponde a cerca de 29 bilhões

de *emails* diários (FOSSI et al., 2011). A Figura 1.1 mostra a porcentagem de *emails* spam ao longo dos anos.

Figura 1.1 – Evolução do percentual de email spam



Fonte: https://www.symantec.com/

Por ser um serviço de baixo custo, o *spam* causa problemas não somente aos usuários e empresas que utilizam *email* no dia a dia, mas também aos provedores de Internet que têm desperdício de banda e aumento de custos na implementação de tecnologias com o objetivo de detectar os mesmos (OLIVO; SANTIN; OLIVEIRA, 2010).

Os spammers (termo utilizado para caracterizar quem envia spam) fazem uso de diversas técnicas afim de dificultar a detecção de suas mensagens. Tais técnicas vão desde a forma como são enviadas as mensagens até a inserção proposital de informações no corpo da mensagem. O objetivo é burlar os mecanismos de detecção de spam, isto é, de evitar que filtros anti-spam interceptem a mensagem.

O *Sender Policy Framework* (SPF) foi criado para ser uma técnica antispam (SEIGNEUR et al., 2004). SPF é uma tecnologia para combater a falsificação de endereços de retorno dos *emails* (*return-path*). O mecanismo permite:

• Ao administrador de um domínio: definir e publicar uma política SPF, onde são designados os endereços das máquinas autorizadas a enviar mensagens em nome deste domínio; e

• Ao administrador de um serviço de *email*: estabelecer critérios de aceitação de mensagens em função de verificação das políticas SPF publicadas para cada domínio.

O processo de publicação de uma política SPF é independente da implantação de verificação de SPF por parte do MTA (*Mail Transfer Agent*). Estes podem ou não ser feitos em conjunto. Embora tenha se mostrado eficiente no começo, o SPF tornou-se obsoleto uma vez que os *spammers* passaram a publicar seus próprios registros SPF. Protocolos de autenticação de *emails* por si só não são suficientemente adequados para a classificação de spams. Protocolos de autenticação são mais eficientes em casos de *phishing*, que é a tentativa de se passar por um site ou *email* autêntico, ou *email spoofing* que é um artifício usado por *spammers* para falsificar o remetente de uma mensagem de *email*. Com o aumento da implementação de mecanismos de detecção de *emails* textuais, os spammers passaram, principalmente a partir do ano de 2006, a enviar *emails* com o texto inserido em imagens (LI et al., 2012). De acordo com (OLIVO; SANTIN; OLIVEIRA, 2010), naquela época, a quantidade de spam utilizando imagens quadruplicou, representando cerca de 25% a 45% do total de mensagens. Esse fato indica que os spammers compartilham técnicas. A maneira como as utiliza varia de acordo com a época ou ocasião.

A classificação de *emails* de texto tenta discriminar se uma mensagem é ou não spam. Nesta abordagem, o sistema verifica as palavras contidas no corpo do *email* e compara se as mesmas ocorrem com frequência em mensagens de spam. Em geral, faz-se uso de um banco de dados contendo palavras recorrentes nestas mensagens (OLIVO; SANTIN; OLIVEIRA, 2010).

Tal técnica, entretanto, se mostra por vezes ineficaz, visto que os spammers trocam caracteres que identificam as palavras, por outros que mantêm o mesmo sentido. Contudo, confundem o mecanismo classificador já que, se um único caractere é trocado, então a palavra não é considerada a mesma combinação de caracteres. Uma palavra como 'Viagra', por exemplo, pode conter diversas formas, e.g. VIAGR4, V.I.A.G.R.A, Vi/a/gra, etc. A quantidade de variações é espantosa, chegando a trilhões de combinações (OLIVO; SANTIN; OLIVEIRA, 2010).

A criação de novas técnicas para a classificação de spam resultam no desenvolvimento de novas formas de envio, elaboradas pelos *spammers*. Estas podem conter alteração do conteúdo da mensagem inserirem de conteúdos anexados. De um modo geral, algumas metodologias de classificação de spam utilizam técnicas de reconhecimento de padrões, vide (OLIVO; SANTIN; OLIVEIRA, 2010). Estas técnicas têm obtido altas taxas de classificações corretas em

ambiente estacionário, algo em torno de 95% de acerto. Ambientes não-estacionários requerem a adaptação de modelos classificadores através de algoritmos incrementais. A ideia é modificar parâmetros para acompanhar as mudanças.

Neste trabalho serão considerados modelos inteligentes evolutivos baseados em nuvem de dados, *Typicality and Eccentricity Data Analytics* (TEDA), e em grânulos fuzzy de informação, *Fuzzy-Set Based evolving Modeling* (FBeM). além disso, o método *Ant Colony Optimization* (ACO) foi implementado a fim de reduzir a dimensionalidade das características.

TEDA é um método que se baseia nos conceitos de excentricidade e tipicidade que representam a densidade e proximidade de dados da estimativa da densidade recursiva. Este método tem por objetivo generalizar e evitar suposições restritivas, geralmente herdadas de métodos estatísticos tradicionais e teoria da probabilidade, tais como independência de amostras de dados, incapacidade de trabalhar com grandes conjuntos de dados e outras suposições de métodos de distribuição de dados. Além disso, abordagens estatísticas tradicionais são geralmente utilizadas em processos aleatórios, contudo, podem ignorar a dependência dos dados em processos reais como clima, economia, biologia, sociologia e psicologia. Estes dados, em sua maioria, são complexos, incertos e pouco conhecidos, porém, não são totalmente aleatórios, como no caso do spam onde é possível, até certo momento, determinar sua estrutura e fonte geradora. Assim como FBeM, TEDA é uma metodologia sistemática que não necessita de dados anteriores e ainda pode ser utilizado para vários métodos de processamento de imagens, clusterização, classificação, previsão, controle, filtragem, regressão, entre outros. Portanto, TEDA se apresenta como uma abordagem estatística alternativa, capaz de trabalhar com qualquer tipo de processos com dados aleatórios, onde cada observação é completamente independente da outra. O spam possui tal característica, uma vez que pode haver várias fontes geradoras do fluxo de mensagens. E ainda, tal problema se torna cada vez mais difícil, na medida em que cada fonte pode utilizar uma ou várias técnicas diferentes. Aliás, novas técnicas podem surgir a cada momento, portanto, é esperado que as abordagens utilizadas para este fim, possam se adaptar sem intervenção de um especialista. FBeM é um método de computação granular que utiliza a informação do tipo fuzzy para construir mapas granulares que associam os espaços de entrada e saída levando em conta que os dados podem ser valores numéricos ou valores incertos. Grânulos consistem de conjuntos de objetos com características semelhantes como equivalência, proximidade, funcionalidade ou indistinguibilidade Modelos baseados no FBeM são criados e evoluem conforme os dados chegam através do fluxo de dados. Tais fluxos podem ser oriundos de tráfego de dados da

internet, dados de áudio e vídeo, clima, etc. O aprendizado no algoritimo FBeM cria grânulos recursivamente, conforme o fluxo. Eventualmente, o quociente da estrutura granular pode ser otimizado, redefinido, deletado e fundido conforme a necessidade.Além disso, a modelagem granular fuzzy oferece algoritmos e modelos com regras simples descrevendo o significado do mapeamento granular entre os espaços. Portanto, a utilização de conjunto fuzzy para dados incertos é a estratégia utilizada por FBeM para lidar distúrbios, julgamento especialista, opiniões e imprecisões. As regras e grânulos FBeM são criados dinamicamente e adaptadas ao longo do tempo conforme necessidade. O método FBeM, assim como TEDA, não necessitam de conhecimento a priori sobre a o problema, para iniciar sua fase de aprendizagem, isso é importante em fluxos online, onde não se conhecem todas ou parte das características dos dados e é preciso, portanto, aprender do zero.

Inteligência de Enxames é o termo utilizado para se referir aos sistemas evolucionários que estudam o comportamento coletivo dos indivíduos de uma população, e a busca de tais indivíduos para soluções simples em problemas complexos. Um enxame é a generalização de uma coleção estruturada de indivíduos capazes de interagir uns com os outros. O algoritmo de otimização por colônia de formigas ACO uma classe de procedimentos que se baseiam no comportamento de formigas. A ideia principal é a reformulação do problema em um grafo, no qual se procura alcançar um caminho otimizado, visando para alcançar determinado objetivo. Formigas artificiais são utilizadas para percorrer tal grafo e durante a época de treinamento, depositando um feromônio artificial nas bordas e vértices de um grafo. As formigas artificiais deverão escolher o melhor caminho em relação a probabilidades que dependem de trilhas de feromônios que foram previamente colocadas pela colônia. Em ACO para seleção de características, as características selecionadas são representadas através de uma combinação de arcos, nos quais as formigas passaram pelo grafo. Cada formiga na colônia deve visitar todos os nós, pelo menos uma vez. Ou seja, elas iniciam em um nó específico, passam por todos os outros nós e terminam o processo em um determinado nó. Cada nó possui dois arcos conectados entre eles, no qual representa seleção ou exclusão da característica a qual esse nó é assimilado. Portanto, a combinação de arcos atravessados juntos oferece uma representação completa de uma solução candidata. A base proposta para este estudo possui 711 características, portanto, foi testado a redução do espaço dimensional objetivando melhorias na acurácia e tempo de processamento das duas abordagens utilizadas.

1.1 Objetivos

Este trabalho tem como objetivo geral desenvolver modelos classificadores a partir do uso de algoritmos incrementais, isto é, algoritmos dirigidos à fluxos de dados online. Os dados são disponibilizados gradualmente e correspondem a valores de atributos extraídos de *emails*.

Como objetivos específicos deste trabalho tem-se:

• Criar uma base de dados que reflita a atual realidade do tipo *emails* spam.

• Extrair características com baixa correlação entre si a partir de uma base de dados de *emails* spam criada. A redução de dimensionalidade do espaço de entrada é fundamental para o desenvolvimento de modelos evolutivos mais compactos.

• Propor e tentar detectar novas instâncias de *emails*, isto é, *emails* com novos padrões de *spam*.

• Demonstrar que métodos incrementais, ou seja evolutivos, são uma abordagem interessante para classificação de *emails* em ambiente variante no tempo.

• Avaliar a possibilidade de graduar amostras de *emails* em classes (exploração do aspecto de transição gradual entre as fronteiras de decisão providas pelos modelos).

• Comparar os modelos e algoritmos de aprendizado FBeM e TEDA. A comparação é baseada em índices de erro, custo computacional e parcimônia.

• Criar uma base de dados que reflita a atual realidade de emails do tipo

1.2 Contribuições

Este trabalho contribui ao campo de pesquisa de sistemas inteligentes evolutivos, sistemas classificadores de um modo geral e à estudos relacionados à classificação de spam *online*. Atualmente, não existem estudos relacionados à classificação de spam *online* utilizando sistemas adaptativos ao longo do tempo. Dessa forma este trabalho discute novas técnicas para classificação de fluxos de dados online.

Espera se com este trabalho demonstrar a eficiência de algoritmos inteligentes evolutivos diante do desafio de classificação de dados online, onde não é possível determinar as regras definem o processo de antemão. Também não é possível processar vários passos de aprendizado (épocas de treinamento) sobre bases de dados completas.

Adicionalmente este trabalho propõe uma nova base de dados contendo diferentes características de *emails*. Bases de dados *benchmark* na área, como a base *Spambase*, já não refletem mais as dificuldades atuais.

1.3 Organização do Trabalho

Este trabalho está organizado em 6 capítulos, conforme descrito abaixo:

Este capítulo contextualiza e introduz o problema de detecção de spam. São descritos os objetivos gerais e específicos e as contribuições do trabalho.

No capítulo 2 é realizada uma revisão bibliográfica sobre o problema de spam, apontando suas principais características. São detalhadas as técnicas utilizadas usualmente para classificação, de *emails* de texto.

O capítulo 3 introduz os classificadores inteligentes evolutivos considerados para detecção online de spam. Tratam se de modelos classificadores de propósito geral propostos recentemente. Os algoritmos FBeM e TEDA são customizados para o problema.

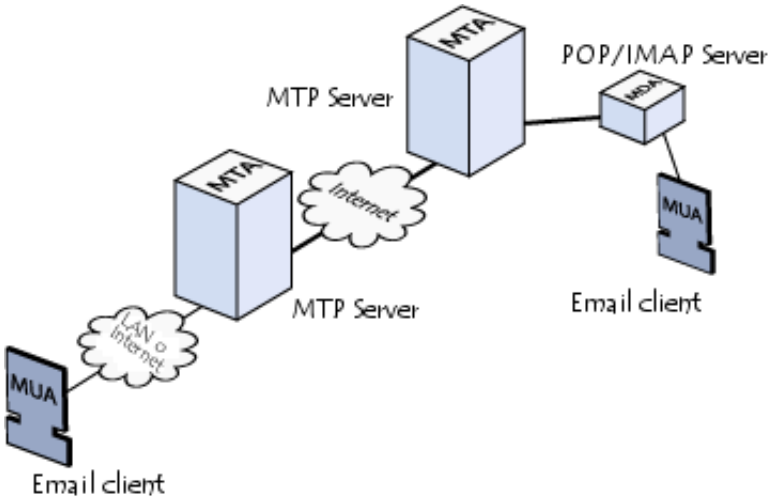
Os capítulos 4, 5 e 6 contém a metodologia empregada nesta pesquisa, os resultados obtidos e a conclusão final, respectivamente.

2 DETECÇÃO DE SPAM

Neste capítulo serão abordadas as principais características de um sistema de envio e recebimento de *email*. Serão apresentadas também, as principais técnicas de envio de mensagens de spam, categorizadas pela maneira como a mensagem é enviada e pelo seu conteúdo. Além disso, alguns métodos para a classificação dessas mensagens serão discutidas afim de apresentar o estado da arte.

O serviço de envio e recebimento de email conta basicamente com dois componentes principais: O *Mail Transfer Agent* (MTA) e o *Mail User Agent*(MUA). O MTA (e.g. *Postfix*) é responsável pelo envio e recebimento da mensagem através de um servidor. O MUA (e.g *Microsoft Outlook*) é responsável por coletar o email do remetente e encaminhar a mensagem utilizando o protocolo SMTP para o MTA de origem. Este se encarrega de enviar a mensagem ao servidor MTA destino. O MTA do destinatário fica responsável por entregar o email para o agente MUA do destinatário utilizando algum protocolo de entrega (e.g. IMAP ou POP). A Figura 2.1 ilustra o processo de troca de mensagens entre dois MTAs.

Figura 2.1 – Troca de mensagens entre dois servidores MTAs.



Fonte: Elaborado pelo autor

O protocolo responsável por definir regras na troca de mensagens entre MTAs é a RFC 2821 (KLENSIN, 2008). Este protocolo especifica essencialmente que o *email* é composto por um cabeçalho, um envelope (*envelope*) e pelo conteúdo (*content*).

O envelope possui o endereço de origem, que é o local para onde relatórios de erros são retornados caso haja endereços destinatários errados.

O conteúdo, por sua vez, é dividido em duas partes, o cabeçalho (*header*) e o corpo (*body*) da mensagem. O cabeçalho contém informações obrigatórias, tais como endereço de *email* do remetente *FROM* e destinatário *TO*, data *DATE*, assunto *SUBJECT* e *email* de copias de mensagem *CC* (*carbon copy*).

O corpo do *email* pode ser composto por texto puro ou, com o uso dos recursos de hipertexto, o H*Hyper Text Markup Language*, (HTML), que é o padrão mais utilizado, também pode conter imagens, o conteúdo será interpretado pelo MUA e apresentado ao destinatário. O cabeçalho da mensagem é codificada no formato *American Standard Code for Information Interchange* (US-ASCII). Já o corpo da mensagem utiliza o padrão *Multipurpose Internet Mail Extensions*(MIME) (FREED; BORENSTEIN, 1996).

A necessidade de adoção do padrão MIME foi devido ao fato de o padrão anterior, *Standard for the Format of ARPA Internet Text Messages* - RFC 822 de 1982, especificar apenas mensagens de texto. Mensagens contendo multimídia, como áudio, vídeo e imagens não eram especificadas neste padrão. Além disso, o padrão RFC 822 é inadequado para usuários que utilizam idiomas que necessitam de caracteres especiais, dos quais a tabela US-ASCCII não suporta (CROCKER, 1982).

O padrão MIME permite:

- Tabela de caracteres para informação textual mais ampla, diferente do US-ASCCII.
- Conteúdo de mensagens divididas em duas ou mais partes (*multi-part*).
- Conteúdo da mensagem com conjunto de caractéres mais amplos.
- Conjunto de formatos diferentes e extensíveis para mensagens não textuais.

O uso de diversos tipos de caracteres, somado à possibilidade do uso do hipertexto e de imagens, permite que o conteúdo da mensagem possa ser explorado na disseminação de spam. Spammers fazem uso de várias técnicas, e utilizam de subterfúgios técnicos para burlar mecanismos antispam.

Por essa razão, os mecanismos antispam precisam se adaptar constantemente às novas técnicas de disseminação utilizadas pelos *spammers*. Um sistema que não consegue se adaptar a estas mudanças tende a se tornar obsoleto.

2.1 Técnicas de envio de *Spam*

A maneira pela qual os *spammers* tiram proveito para difundir o spam passou por grandes mudanças nos últimos anos. Com o desenvolvimento de novas técnicas de detecção, cada

vez mais eficazes, os *spammers* se viram obrigados a aprimorar suas técnicas a fim de enganar os sistemas. Podemos classificar as técnicas de envio de *spam* em duas categorias principais: i) técnicas baseadas no envio da mensagem; e ii) técnicas baseadas no conteúdo da mensagem.

As técnicas baseadas no envio da mensagem se referem à forma como a mensagem é enviada (e.g através de um *MTA* legítimo ou um *script* de envio). Já as técnicas baseadas no conteúdo da mensagem utilizam meios de confundir os sistemas de detecção que se baseiam no conteúdo, textual ou não, da mensagem. Há ainda a prática de combinar as duas técnicas ou até mesmo várias técnicas da mesma categoria, aumentando ainda mais a capacidade de o *email spam* não ser detectado pelos mecanismos de classificação.

Tanto os pesquisadores que desenvolvem novas técnicas de classificação de mensagens quanto os administradores de sistemas de *email*, que implantam e configuram os mecanismos antispam, estão sujeitos a grandes desafios técnicos. Uma vez que existe uma grande variação de técnicas de spam os sistemas precisam estar preparados para evoluir de acordo com as novas investidas e garantir a correta classificação das mensagens.

Serão abordadas a seguir, as duas categorias principais de envio de mensagens de *spam*, cada uma delas, possuem várias subcategorias e serão tratadas a seguir.

2.2 Técnicas Baseadas no Envio da Mensagem

As técnicas que se fundamentam no envio de mensagens tratam da forma como a mensagem é enviada para o destinatário. Em outras palavras, essas técnicas se preocupam com o *email* é "disparado", e não com o seu conteúdo. Os *spammers*, por exemplo, podem utilizar um MTA autêntico, de maneira fraudulenta, e utilizá-lo com o objetivo de enviar *spams* ainda com a possibilidade da verificação do recebimento da mensagem. Caso essa não tenha sido recebida, o sistema pode enviar outras sucessivamente. Outras técnicas baseadas no envio de mensagens serão tratadas a seguir, nos seguintes itens:

a) **Mecanismo de envio simples**

Esse tipo de técnica tem o objetivo de atingir o maior número possível de destinatários, no menor tempo possível, enviando milhares de mensagens sem se preocupar com erros ou bloqueios de servidores. A configuração padrão de um servidor MTA permite verificar se a mensagem de *email* foi recebida corretamente isso por que existem vários problemas que podem ocorrer no envio da mensagem como; destinatário inexistente cota de disco do destinatário esgotada, erros de rede, etc. Em alguns casos, o MTA coloca a mensagem em uma fila para

uma nova tentativa de envio, essa nova tentativa varia de acordo com a configuração do MTA. Tais mecanismos, configurados para garantir a confiabilidade do serviço e a redução de erros, demandam auto custo computacional uma vez que utilizam complexos softwares de envio, segurança e monitoramento. A Figura 2.2 demonstra um exemplo de MTA configurado conforme mencionado acima.

Figura 2.2 – Troca de mensagens entre dois servidores MTAs.

Evento	Remetente	Hora do envio	Pontuação de spam	Destinatário	Resultado	Ações
⚠	adalgisa@cemes.edu.br	20/06/2016 14:28:14	0	rcradm@hotmail.com	SMTP error from remote mail server after RCPT TO: <rcradm@hotmail.com>: 452 4.5.3 Error: too many recipients	ⓘ
✓	adalgisa@cemes.edu.br	20/06/2016 14:28:14	0	edilson.rodrigues22@gmail.com	Aceito	ⓘ
✓	adalgisa@cemes.edu.br	20/06/2016 14:28:14	0	mpaulaacaa@uol.com.br	Aceito	ⓘ
⚠	adalgisa@cemes.edu.br	20/06/2016 14:28:14	0	sinhana.bh@terra.com.br	Connection timed out	ⓘ
✓	adalgisa@cemes.edu.br	20/06/2016 14:28:14	0	brunotopes28@hotmail.com	Aceito	ⓘ
✗	adalgisa@cemes.edu.br	20/06/2016 14:28:14	0	hugogondim@yahoo.com.br	retry timeout exceeded	ⓘ
⚠	adalgisa@cemes.edu.br	20/06/2016 14:28:14	0	eng.rodrigobahia@hotmail.com	SMTP error from remote mail server after RCPT TO: <eng.rodrigobahia@hotmail.com>: 452 4.5.3 Error: too many recipients	ⓘ

Fonte: Elaborado pelo autor

Uma vez que o MTA não esteja configurado para garantir a confiabilidade do serviço, ou seja, não se preocupa com o recebimento da mensagem e nem com o tratamento de erros, o mecanismo de envio da mensagem se torna muito mais simples, não necessitando de softwares complexos de envio, segurança e monitoramento, demandando assim, menor custo computacional.

O disparo massivo de mensagens pode ocorrer de forma simples e automática, um software de envio pode ser empregado para a configuração das mensagens e o procedimento não necessita de conhecimentos em programação, dessa forma o *spammer* consegue enviar milhares de mensagens em intervalos regulares ou ao mesmo tempo.

Quando um *spammer* envia uma mensagem para uma conta de *email* que não existe, o servidor SMTP do destinatário retorna uma mensagem de erro relatando que o endereço de *email* não foi encontrado, porém, como o servidor de envio não está configurado para tratar esse erro, o mesmo é descartado. A lista dos códigos relativos aos erros pode ser encontrada na RFC 3463[36].

b) **Envio de mensagem com tratamento de erro**

Ao contrário do envio simples, a técnica de envio de mensagem com tratamento de erro, tem por objetivo garantir ao máximo que as mensagens de spam cheguem ao destinatário, os

servidores MTA de envio são configurados para tratarem os erros que os servidores de destino retornam, por exemplo, quando existe um erro de rede temporário ou erro no servidor MTA do destinatário. Se o *email* não chega corretamente ao destinatário e o servidor de destino retorna o erro ao servidor de envio, este último recoloca a mensagem em uma fila de espera para enviar novamente, o intervalo de envio e a quantidade de vezes que o *email* poderá ser recolocado na fila de envio é variável, depende de como o servidor for configurado.

Este tipo de técnica é mais utilizada em relação a anterior, uma vez que, ao se tratar o erro, o *spammer* conseguirá alcançar maior número de destinatários e ainda, poderá excluir da sua base de dados, *email* inexistentes. A Figura 2.3 exemplifica o envio de mensagens com tratamento de erro. O servidor MTU do remetente envia a mensagem para o servidor do destinatário, se o endereço de *email* for inválido, o servidor MTU do destinatário retornará uma mensagem informando que o endereço de *email* não existe. Dessa forma, o servidor do *spammer* ao verificar que o *email* não existe descarta a mensagem e remove o endereço de *email* de sua base de dados.

Figura 2.3 – Envio de mensagem com tratamento de erro entre MTAs.

Fonte: Elaborado pelo autor

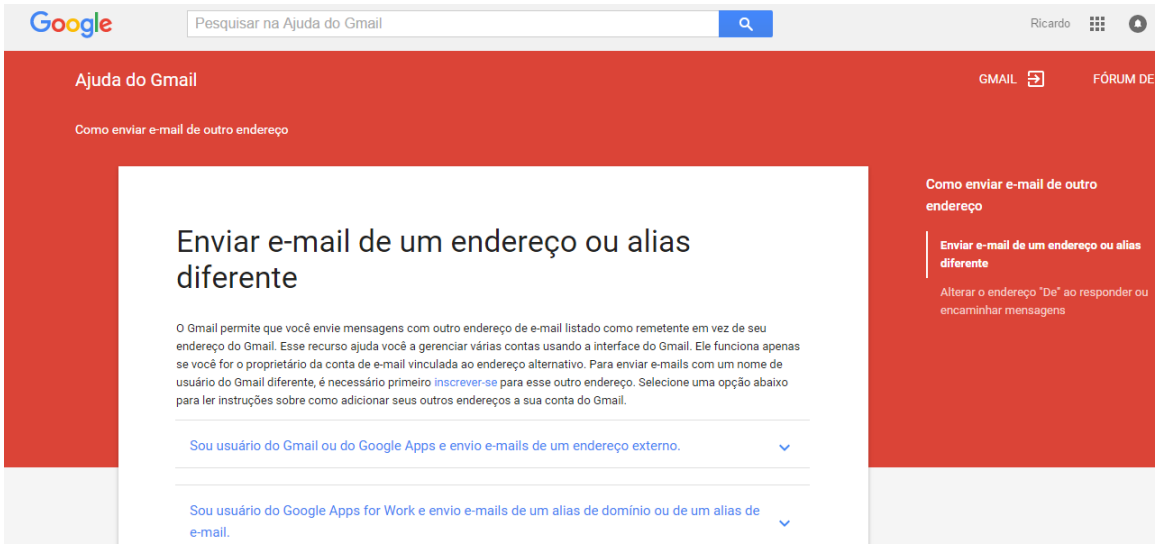
Esta técnica de envio não é exclusiva aos *spammers*. Empresas de *marketing online* adotam o mesmo procedimento de envio, uma vez que é interessante que o maior número de pessoas receba as mensagens. É importante se atentar aos erros para evitar bloqueios antispam relacionados a erros em mensagens com grande número de destinatários.

Esta técnica de envio também gera dificuldades nos mecanismos de classificação de *spam*, pois a maneira como é enviado a mensagem não se difere da forma como os *spammers* enviam.

c) **Envio com substituição de endereço do remetente ou servidor de transmissão**

O protocolo de envio de *email*, SMTP, possui em sua arquitetura brechas que são amplamente utilizadas pelos *spammers* (KLENSIN, 2008), como a possibilidade de trocar o endereço de *email* do remetente por um *email* diferente do de quem está enviando a mensagem. Assim, utiliza se um servidor em um domínio *x* para enviar um *email* de um domínio *y*, causando transtornos para os usuários pois os mesmos podem facilmente confundir a mensagem como sendo de um remetente confiável. Essa prática não é sobretudo uma fraude, vários serviços de *email* possibilitam que seus usuários enviem mensagens através de seus servidores utilizando endereços de *email* diferentes, como mostrado na Figura 2.4.

Figura 2.4 – Ajuda do *Gmail* para enviar *email* de um endereço diferente



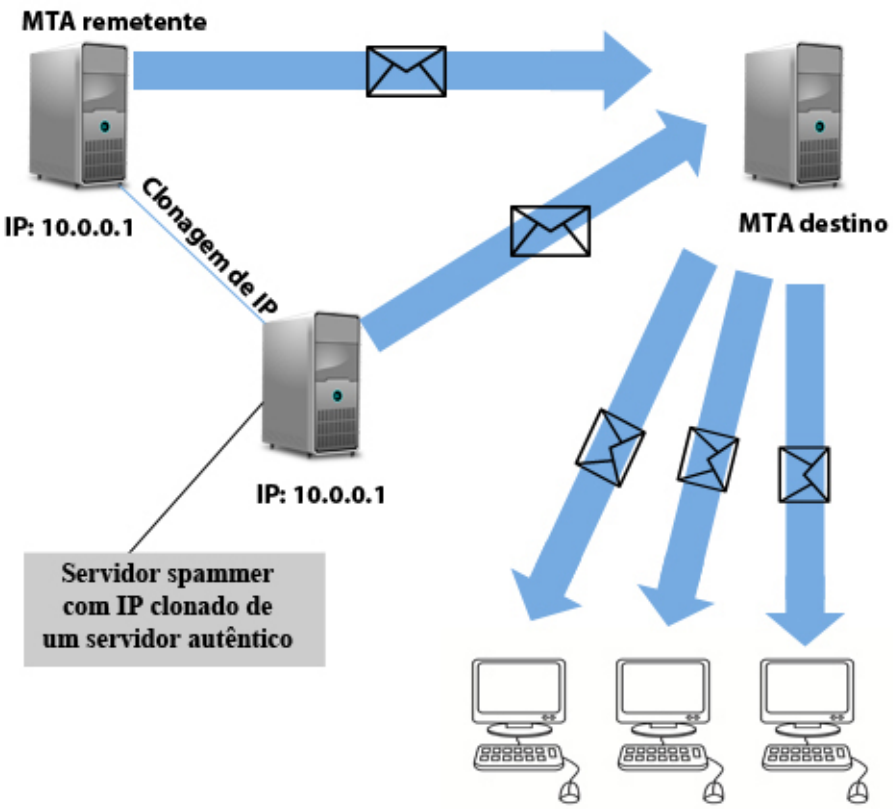
Fonte: https: <//support.google.com/mail/answer/22370?hl=pt-BR>

Além disso, o spammer poderá combinar outra técnica, clonando o endereço de IP (*internet protocol*) do MTU remetente, poderá burlar os sistemas antispam que se baseiam no endereço de IP do remetente, como mostrado na Figura 2.5.

Essa prática é muito utilizada por *spammers* que praticam *phishing*. O servidor do destinatário reconhece o servidor de *spam* como autêntico. O conteúdo da mensagem poderá conter *links* que redirecionam para o servidor *spam*, e neste, poderá estar configurado um servidor *Web* que contém páginas que imitam um ou vários *sites* autênticos, seja de um banco, cartão de crédito, rede social etc. Ao ler a mensagem, o usuário poderá ser facilmente convencido de que se trata de uma comunicação autêntica e poderá ter seus dados comprometidos além de prejuízos morais e financeiros.

d) **Envio através de scripts maliciosos**

Figura 2.5 – Clonagem de *IP* de um servidor autêntico por um servidor *spammer*

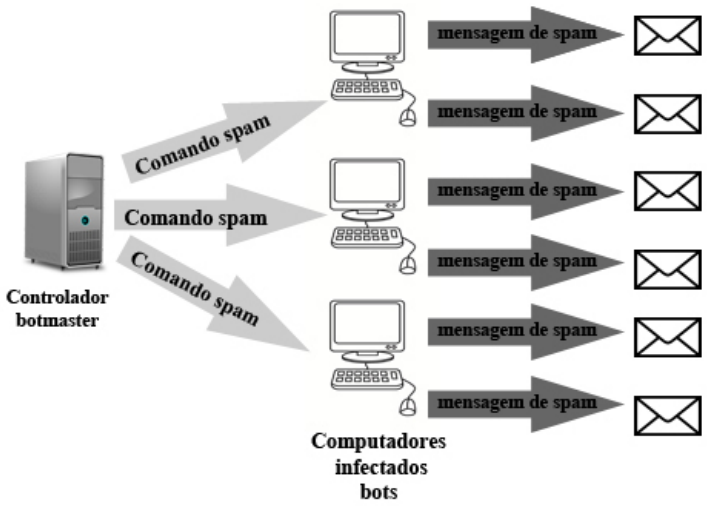


Fonte: Elaborado pelo autor

Um *script* malicioso para envio de spam é basicamente um código compilado para iniciar de maneira remota. Em geral, esse tipo de código contamina o computador do usuário através de uma mensagem de *spam*, softwares piratas, vírus de computador, pela rede ou através de dispositivos externos como o *pendrive*. Ao contaminar o computador do usuário, o *script* envia para um endereço pré estabelecido uma mensagem contendo informações sobre o estado da máquina da vitima. Quando o *spammer* decide enviar as mensagens de *spam*, ele manda um comando para o computador da vítima ou computadores. A partir daí o computador do usuário começará a enviar mensagens de *spam* com o conteúdo que o *spammer* estabelecer sem que o usuário saiba. Os computadores infectados são conhecidos como *bots* e o atacante ou servidor remoto como *botmaster*. A Figura 2.6 ilustra esta técnica.

Essa técnica de envio de *spam* pode causar congestionamento nos servidores destinos uma vez que o número de computadores infectados pode ser extremamente grande. Além disso, os filtros antispam terão dificuldade em bloquear a fonte de envio, pois os computadores infectados podem estar em diferentes países ou continentes. Dessa forma, também é extremamente difícil localizar o endereço do *botmaster*, o que facilita, seu anonimato. Dependendo do número

Figura 2.6 – Envio de *spam* através de *script* malicioso, controlado remotamente



Fonte: Elaborado pelo autor

de computadores infectados, somado a grande variedade de faixa de *IPs* disponíveis e ainda, devido ao uso de outras técnicas como servidores *proxys*, os sistemas que classificam as mensagens de *spam* baseado no endereço de envio são ineficazes. Eles não conseguirão estabelecer um bloqueio das mensagens se estas têm origens diferentes.

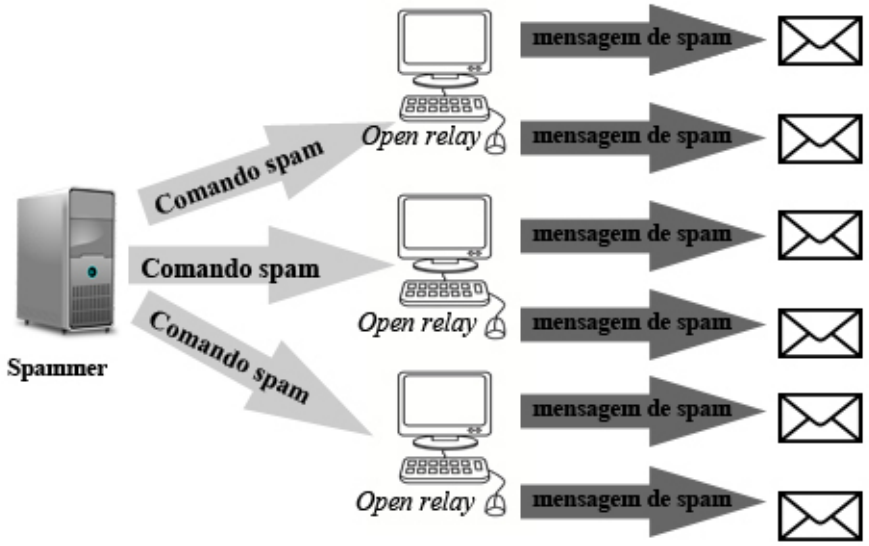
O envio de *spam* através de *scripts* maliciosos é parecido com mecanismos simples de envio uma vez que não há custo computacional para o atacante. O custo total é dividido entre os computadores infectados. Além disso, não há custos financeiros para enviar o comando remoto ou manter os computadores infectados.

e) **Envio através de servidores de retransmissão abertos (*open relays*)**

Um servidor de retransmissão aberto, conhecido como *open relay* é um servidor SMTP onde qualquer pessoa poderá se conectar e enviar mensagens de maneira irrestrita sem a necessidade de autenticação. As mensagens são enviadas, na maioria dos casos, sem o conhecimento da empresa responsável pelo servidor. No início da Web, na década de 90, e até meados do ano 2000, era comum a pratica de envio de *spam* utilizando tais servidores. Nessa época, eram abundantes em toda a Internet. Alguns computadores vinham pré configurados para serem um servidor do tipo *open relay*. Contudo, ao verificar quantidade de *spams* que estavam sendo enviados através desses servidores, os desenvolvedores de *MTAs* passaram a configurar seus servidores para serem um *closed relay*, ou seja, os servidores somente enviariam mensagens de usuários autorizados e autenticados (BIANCHI, 2015). A técnica de envio utilizando *open*

relays é semelhante à de envio utilizando *scrips* maliciosos (*botnets*), conforme ilustra a Figura 2.7 .

Figura 2.7 – Envio de *spam* através de *script* malicioso, controlado remotamente



Fonte: Elaborado pelo autor

Apesar de a prática de envio de spam utilizando servidores abertos ter diminuído, devido as configurações impostas pelos desenvolvedores de MTAs e também devido a novas técnicas de segurança implementadas em servidores, em 2012 e 2013 foram registrados cerca de quatro mil novos servidores *open relays* e ainda, todos os dias cerca de 10 a 20 novos servidores *open relays* são descobertos por *spammers* (BIANCHI, 2015).

Neste caso, o bloqueio por endereço de IP de origem pode ser ineficaz, visto que todos os dias são encontrados novos servidores abertos e ainda, um servidor continuará bloqueado mesmo depois de ser devidamente configurado. Isto causa bloqueio indevido de *email* legítimos.

Nesta seção foram apresentadas as principais técnicas de envio de *spam* baseadas no envio, tais técnicas não se preocupam com o conteúdo da mensagem, ou seja, não se preocupam em mascarar o conteúdo a fim de não serem classificados por um sistema antispam. A maneira como é enviada a mensagem é a preocupação central dos *spammers* que utilizam tais técnicas.

2.3 Técnicas Baseadas no Conteúdo da Mensagem

Diferentemente das técnicas de envio de mensagens baseado na forma de envio, as técnicas de envio de *spam* baseadas no conteúdo do *email* estão relacionadas à formas de burlar os mecanismos antispam que utilizam se de informações extraídas do corpo do *email* para realizar

a classificação da mensagem. Tais técnicas vão desde a inserção de palavras que podem confundir os sistemas classificadores até ao uso de técnicas de programação utilizando recursos da linguagem HTML, que pode ser incorporada ao *email*.

A maneira como será apresentado tais técnicas utilizará da mesma sequência que foram apresentadas as técnicas baseadas na forma de envio, na seção anterior, sendo apresentados os principais métodos nos itens entre *a* e *f*.

a) **Inserção proposital de palavras**

A técnica de *spam* utilizando a inserção proposital de palavras tem por objetivo confundir os sistemas classificadores de *spam*, tais classificadores verificam a ocorrência de palavras e, baseado na ocorrência das mesmas, classificam a mensagem. Os classificadores *bayesianos*, classificam os *email* verificando a ocorrência de determinadas palavras no corpo do mesmo. Dependendo da quantidade de palavras, o classificador avalia probabilisticamente se o *email* é *spam* ou não.

Para confundir esse tipo de classificador, o *spammer* insere palavras de maneira proposital oriundas de bases contendo *email* autênticos, no corpo da mensagem, com o objetivo de aumentar a predominância de textos legítimos, ao verificar a ocorrência de palavras, o classificador retornará maior probabilidade de que a mensagem não seja *spam*.

b) **Ofuscação textual**

As técnicas de ofuscação textual, também conhecidas como troca ou inserção de caracteres é utilizada para burlar os sistemas de detecção que se baseiam no texto da mensagem. Diferente da técnica de inserção proposital de palavras, que visa inserir palavras a fim de diminuir a probabilidade estatística de palavras relacionadas a *spam*, a ofuscação textual insere, altera ou exclui caracteres em palavras mais comuns, consideradas indícios de *spam*.

Os sistemas que utilizam probabilidade de palavras e ou se baseiam em detecção por palavras chaves encontram grande dificuldade nas mensagens que utilizam ofuscação textual, uma vez que uma palavra pode ser escrita de diversas maneiras, não perdendo o seu real significado. Uma palavra como 'Viagra', por exemplo, pode conter diversas formas (e.g. VIAGR4, V.I.A.G.R.A, Vi/a/gra, etc), a quantidade de variações é abissal, chegando a cerca de 600426974379824381952 combinações possíveis (COCKERHAM, 2004).

A Figura 2.8 exemplifica algumas variações entre caracteres e símbolos que podem ofuscar uma palavra como, por exemplo, viagra, presente em muitas mensagens de *spam*.

Figura 2.8 – Variações de símbolos e caracteres

V	I	A	G	R	A
V, v, ∨	I ı 1 l | ɪ i : Ì Í Î Ï	A a @ ^ á à â ã ä å æ Á À Â Ã Ä Å	G g	R r ®	A a @ ^ á à â ã ä å æ Á À Â Ã Ä Å
3 variations	12 variations	17 variations	2 variations	3 variations	17 variations

Fonte: http://cockeyed.com/lessons/viagra/viagra.html

Devido a enorme quantidade de possibilidades de combinações de cada palavra, os filtros baseados na ocorrência de palavras, mesmo que verificassem variações das mesmas, teriam um custo computacional muito alto já que seria necessária a verificação palavra por palavra considerando todas as combinações possíveis. A quantidade de caracteres presentes no padrão *Unicode* 8.0 é de 120672, sendo capaz de representar códigos representativos de caracteres de vários idiomas, ideogramas e símbolos (ALLEN et al., 2007).

A dificuldade de classificar uma mensagem contendo ofuscação de caractere pode ser ainda maior quando o *spammer* troca um caractere por dois ou mais caracteres, dessa forma, o número de combinações cresce exponencialmente.

Um experimento realizado por (LIU; STAMM, 2007) demonstrou o quanto a ofuscação textual influência no resultado de um classificador. No experimento, foram realizadas substituições de caracteres de uma base sem ofuscação por caracteres que possuíam semelhança visual com o caractere original. Em seguida consideraram o *software SpamAssassin*, uma das ferramentas mais utilizadas em servidores SMTP para classificação de *email*, esta ferramenta atribui uma pontuação para a mensagem onde, quanto mais alta a pontuação, maior a possibilidade da mensagem ser um *spam*. Foram feitos testes de classificação nas bases originais sem ofuscação, nas bases ofuscadas e desofuscadas. Em um dos testes, as mensagens originais de *spam* receberam uma pontuação de 7,9 a 21,7. Porém, ao testar a base com ofuscação, as pontuações foram de 1,9 a 5,94, demonstrando assim que tal técnica pode dificultar um sistema classificador que leva em conta a variabilidade de palavras e ou caracteres.

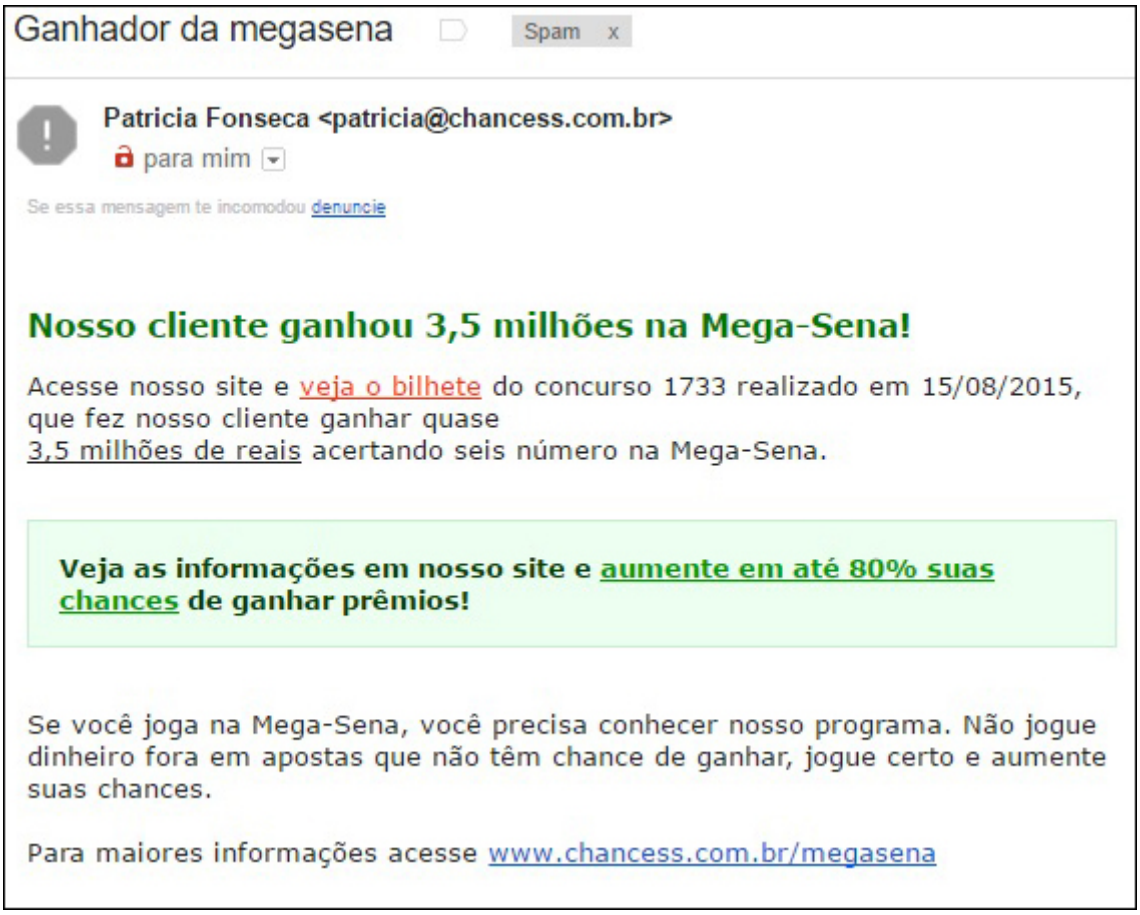
c) **Engenharia Social**

No contexto de segurança da informação, a engenharia social refere se a manipulação psicológica de pessoas com objetivo de realizar ações ou fornecer informações pessoais. Este termo está relacionado também, com um tipo de intrusão psicotécnica que depende profundamente da iteração humana e busca enganar pessoas para quebrar padrões de segurança.

Os *spammers* que disseminam o *phishing* utilizam a técnica de engenharia social a fim de ludibriar o usuário a cometer alguma ação que possibilitará a aplicação de algum tipo de golpe, ou até mesmo no fornecimento de dados confidenciais.

Na Figura 2.9 pode ser observada a tentativa de enganar o usuário com uma mensagem oferecendo vantagens financeiras e métodos que prometem aumentar as chances em jogos de loteria, a mensagem contém um *link* que, ao acessado, o mesmo poderá levar o usuário a um *site* de *phishing* ou com conteúdo de códigos maliciosos.

Figura 2.9 – Mensagem de spam utilizando engenharia social



Fonte: Elaborado pelo autor

Este tipo de mensagem utiliza um domínio válido na internet e não utiliza técnica de ofuscamento de caracteres ou palavras, como pode ser observado pela Figura 2.9. Além disso, este tipo de mensagem possui características muito parecidas com *emails* legítimos, o que favorece ainda mais o atacante, tornando o *email* convincente aos olhos de alguns usuários.

d) **Conteúdo em imagens**

Ao longo do tempo, muitos métodos antispam foram desenvolvidos para reconhecimento de textos em mensagens. Contudo, os spammers também evoluíram suas técnicas e, uma vez que, a maioria dos classificadores buscavam somente padrões em textos, passaram a incorporar imagens nas mensagens (LI et al., 2012). Em função do fato de que é possível incor-

porar HTML nos *email*, os *spammers* começaram a inserir texto nas imagens com o objetivo de driblar os filtros de texto.

Uma das soluções para contornar este problema foi a utilização de técnicas de OCR (*Optical Charecter Recognittion*) - Reconhecimento Ótico de caracteres. Esse tipo de técnica busca reconhecer e extrair textos contidos em imagens. Diante disso, os *spammers* alteraram a forma como desenvolviam as imagens, utilizando técnicas de ofuscação com o objetivo de não serem detectados, ao alterar o fundo das imagens e o formato dos caracteres inseridos, conseguiram dificultar os sistemas de OCR.

Segundo (LI et al., 2012), este tipo de técnica passou a ser mais utilizada a partir de 2006. Naquele ano a quantidade de *email* utilizando esta técnica representou cerca de 25 a 45% do total de mensagens. Além dos transtornos causados aos usuários, este tipo de mensagem pode congestionar facilmente o MTA do destinatário, uma vez que o tamanho de uma mensagem com imagem é superior a outra que só possui texto.

e) **Incorporação de HTML nas mensagens**

O uso do HTML nas mensagens de *email* é amplamente utilizado pelos *spammers* para disseminar suas mensagens. É possível esconder textos, inserir imagens, *links* e scripts possibilitarão maior dificuldade para os filtros antispam.

Utilizando o recurso de *hiperlink* do HTML é possível exibir uma mensagem com um link oculto que levará o usuário a um site diferente do que se propõe o texto e ainda, exibir uma mensagem de um link válido quando o usuário posicionar o mouse sobre o mesmo, por exemplo:

<a href="http://www.sitespam.com"title="http://www.caixa.gov.br"clique aqui para acessar o site do seu banco

No exemplo acima, o usuário ao posicionar o *mouse* sobre o *link* contido na frase "clique aqui"verá uma caixa de texto com o endereço de *link* de um *site* autêntico <http://www.caixa.gov.br>. Porém ao clicar no link, será redirecionado para outro endereço <http://www.sitespam.com> como ilustra a Figura 2.10.

Figura 2.10 – Mensagem com endereço de link oculta

Fonte: Elaborado pelo autor

No trabalho (OLIVO; SANTIN; OLIVEIRA, 2010), há vários exemplos de outros casos de uso do HTML em mensagens de spam. Outras técnicas com o uso de HTML são:

Comentários HTML: Os *spammers* camuflam o conteúdo de suas mensagens usando comentários HTML ou outras *tags* que serão, por padrão, removidas pelo gerenciador de *email* do usuário e mostrará a mensagem de *spam*. Dessa forma, as palavras são exibidas e o filtro é enganado.

Mensagem bilíngue: Uma mensagem legítima é gerada e enviada ao destinatário. Porém, através do HTML, é gerado outra mensagem com conteúdo *spam*, dessa forma, o gerenciador de correio exibe, por padrão, a segunda mensagem. O filtro antispam analisa as duas mensagens, contudo, classifica o *email* como autêntico.

Tags HTML inválidos: Os *spammers* geram *tags* de HTML inválidas, com o objetivo de fazer com que os textos considerados legítimos sejam analisados pelo filtro, porém, a mensagem de *spam* é exibida no corpo do *email*.

Texto redundante: A mensagem é enviada contendo uma parte com conteúdo legítimo e outra, codificada em HTML, o conteúdo *spam*. O gerenciador de *email* do usuário exibe a mensagem com o código HTML (OLIVO; SANTIN; OLIVEIRA, 2010).

f) **Peças publicitárias (*email marketing*)**

Peças ou campanhas publicitárias conhecidas como *email marketing* são *emails* contendo mensagens de empresas com a finalidade de divulgação de algum produto ou serviço, em geral as empresas contratam provedores que fornecem o serviço de envio em massa de mensagens com tratamento de erros e oferecem orientações para que as mensagens não sejam classificadas como *spam*.

É importante ressalvar que o *email marketing* nem sempre é utilizado com a finalidade de disseminar *spam*, as empresas que utilizam este serviço constroem suas bases de endereços oferecendo alguma vantagem ao cliente, dessa forma, o cliente pode receber descontos, novidades e promoções diretamente em seu *email* com a possibilidade de cancelar a assinatura do recebimento deste tipo de mensagem.

Muitas das vezes os usuários classificam essas mensagens como *spam* por não gostarem de recebê-las diariamente em suas caixas de entrada, preferindo classifica-las como *spam* do que cancelar a assinatura do serviço. Contudo, *spammers* utilizam esse método para envio de spam com a finalidade de se passar por empresas ou para promover venda de produtos, como por exemplo medicamentos controlados (e.g Viagra).

Nesta seção foram apresentadas as principais técnicas de envio de *spam* baseadas no conteúdo da mensagem, esse tipo de abordagem se preocupa no conteúdo inserido na mensagem e não da forma como ela é enviada. As técnicas utilizadas pelos *spammers* podem utilizar um ou mais métodos de envio, quanto maior a quantidade de métodos empregados nas mensagens mais difícil será para que os sistemas classificadores possam identificar corretamente a existência de *spam* nas mensagens.

2.4 Técnicas para Classificação de *Spam*

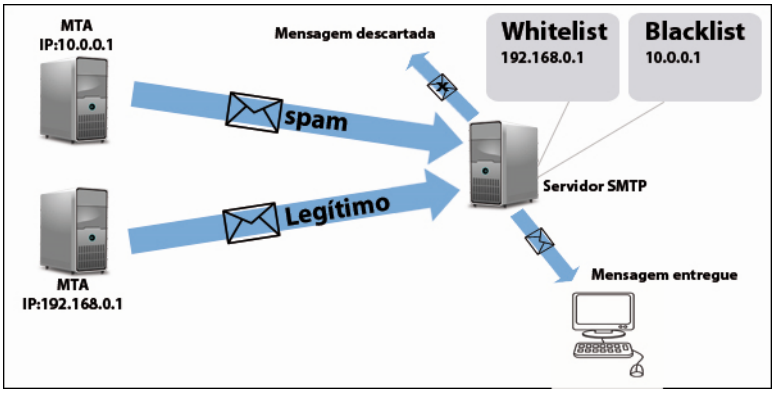
Nesta seção serão apresentadas as principais técnicas de classificação de *spam* encontradas na literatura. Tais técnicas utilizam listas de bloqueio, palavras chave, recusa intencional, políticas de remetente, técnicas baseadas em assinaturas e reconhecimento de padrões.

a) **Listas de bloqueio (*whitelists blacklists*)**

Uma das técnicas mais utilizadas para interceptar *spams* é o uso de listas de bloqueio. Tal método consiste em bloquear mensagens provindas de endereços de IPs contidos em listas negras ou *blacklists*, que são alimentadas por provedores do mundo inteiro e liberar mensagens que se encontram em listas brancas ou *whitelists*.

Em ambos os casos, há a possibilidade de inserção ou retirada do endereço do remetente de uma das listas. Por exemplo, se um remetente começar enviar spam, seja por ataque de um agente externo ou por iniciativa própria, seu endereço de IP poderá ser inserido em uma lista negra mediante a denúncia de algum provedor de serviço de *email* ou Internet. Usuários também podem reportar mensagens de *spam*, porém, somente aos provedores de *email*.

Figura 2.11 – Ilustração de um servidor SMTP com lista de bloqueio



Fonte: Elaborado pelo autor

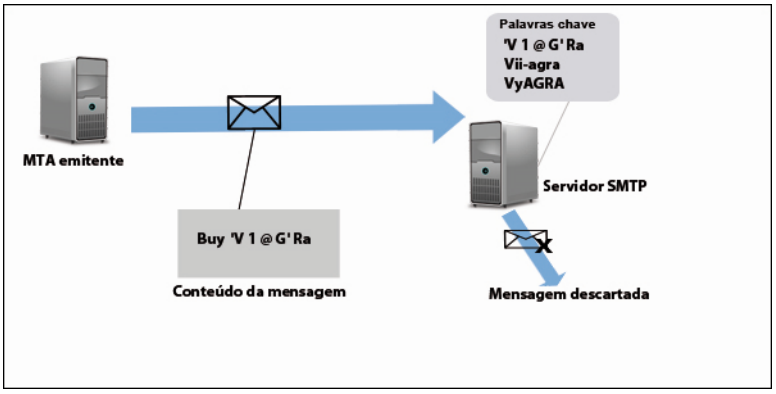
Na Figura 2.11, o servidor SMTP, ao receber uma mensagem, verifica se o endereço de origem está listado em uma *blacklist* ou *whitelist*, no primeiro caso o *email* é descartado, no segundo, a mensagem é entregue ao destinatário.

Contudo a adoção deste tipo de método pode gerar problemas de segurança, spammers podem utilizar métodos de clonagem de endereços IPs e com isso passarem despercebidos pelos servidores SMTP, além disso, se um atacante conseguir invadir um servidor legítimo, este poderá enviar mensagens de *spam* sem ser descoberto. Além do mais, se o emissor utiliza um serviço de envio compartilhado (e.g. *Yahoo*, *Gmail*) e for inserido em uma *blacklist*, usuários legítimos ficarão prejudicados ao ter suas mensagens classificadas como *spam*.

b) **Classificação por palavras chaves**

Algumas abordagens classificam *email* através da existência ou não de palavras chaves, neste método uma lista contendo determinadas sequências de caracteres é consultada sempre que um servidor SMTP recebe uma mensagem, caso o *email* contenha palavras presentes nesta lista, a mensagem é classificada como spam. A Figura 2.12 ilustra um servidor SMTP consultando o conteúdo de uma mensagem e classificando a mesma como spam devido conter sequências de caracteres presentes na lista.

Figura 2.12 – Ilustração de um servidor SMTP com lista de bloqueio



Fonte: Elaborado pelo autor

A classificação de *spam* utilizando palavras chave deve ser utilizada com bastante cautela uma vez que quanto maior o número de sequência de caracteres presentes na lista de verificação, maior será o custo computacional para verificar a mensagem, além disso, spammers poderão inserir textos em imagens ou fazer uso de alguma técnica HTML para burlar este tipo de sistema.

c) ***Graylisting***

A técnica de *graylisting* consiste em recusar, inicialmente, uma mensagem de *email*, colocando-a em uma lista, temporariamente, e retornar uma mensagem de falha de entrega ao MTA do emitente informando que a mensagem foi colocada na *graylist*. Como há muitas mensagens que partem de *botnets* e de servidores de envio mal configurados, ao receber a mensagem o servidor não reprocessará a mensagem, ou seja, não enviará novamente a mensagem ao destinatário. Caso o servidor envie a mensagem novamente, o MTA do destinatário consultará se a mesma consta na *graylist*. Para o caso positivo, a mesma será encaminhada ao destinatário.

(LEVINE, 2005) lista algumas situações em que a utilização de graylists pode causar falha. Entre elas estão: (*i*) classificação incorreta de *email* legítimos, caso o MTA de origem não esteja devidamente configurado para reprocessar a mensagem com falha na entrega; (*ii*) Atraso de entrega do *email*, uma vez que a mensagem é colocada em uma *graylist* e enviado uma mensagem para o servidor de envio, até que o mesmo reprocesse a mensagem e envie novamente ao servidor de destino e este, por sua vez, verifique se a mensagem está presente na *graylist*, poderá atrasar consideravelmente o tempo de entrega. Servidores de envio de *spam* devidamente configurados poderão reprocessar a mensagem reenviando para o servidor do destinatário a mesma mensagem, isto inviabiliza o uso da técnica de *graylisting*.

d) ***SPF - Sender Policy Framework***

O *Sender Policy Framework* (SPF) é uma estrutura que estabelece métodos para prevenir que *email* tenham seus endereços de remetente substituídos. Para isso, ele determina que o proprietário de um domínio deve especificar quais servidores estão autorizados a enviar mensagens de *emails*. Registros SPF devem ser cadastrados em servidores *Domain Name System* (DNS). Estes serão consultados posteriormente por servidores SMTP de destino. Caso a autenticidade do emitente não seja confirmada, a mensagem não será encaminhada ao destinatário.

Como as técnicas de spam se aprimoram com o intuito de se posicionarem diante de novos métodos de detecção, boa parte dos spammers consegue publicar registros de SPF uma vez que conseguem ter acesso a MTAs completos. Isso se deve ao fato de que muitos provedores oferecem planos de *email marketing*, que é a utilização do *email* como ferramenta de *marketing* direto, e não estabelecem políticas antispam e não se preocupam em verificar a idoneidade de seus contratantes.

e) **Técnicas de assinatura de mensagens**

Ao longo do tempo, desde o surgimento do *spam*, foram desenvolvidas várias técnicas baseadas em assinaturas na tentativa de aplacar o problema do *spam*. O *Domain Keys Identified*

MAIL (DKIM) propõe métodos pelos quais as mensagens devem ser assinadas digitalmente. Dessa forma, um domínio assinante pode reivindicar a autoria e responsabilidade pela inserção da mensagem no fluxo de dados de *email*. Os destinatários podem averiguar a autenticidade da assinatura solicitando a chave pública diretamente ao emitente e, dessa forma, verificar se a mensagem foi enviada de um domínio válido. O DKIM foi especificado pelo IETF (*Internet Enginnering Task Force*), órgão internacional que cria padrões e definições de autenticação de mensagens baseadas em chaves públicas criptografadas. Este método permite ainda que o conteúdo da mensagem seja também verificado, dessa forma, o método também auxilia no combate ao *phishing*.

Quatro outras técnicas de assinatura foram especificadas pelo IETF com o objetivo de padronizar as mensagens enviadas pela internet, são elas: PEM - (*Privacy Enhaced Mail*) (LINN, 1993); PGP - (*Pretty Good Privacy*) redefinida mais tarde como OpenPGP (CALLAS et al., 2007); MOSS - MIME (*Object Security Services*), uma evolução do PEM (CROCKER, 1982); e S/MIME - (*Secure MIME*), essa, por sua vez, foi desenvolvida pela RSA (*Security*) e mais tarde foi homologada pela IETF(RAMSDELL, 2004).

As técnicas de assinatura de mensagens de *email* são soluções eficazes no combate ao *spam*. Contudo, não existe uma padronização para envio de mensagens pela internet. Além disso, o protocolo SMTP possui brechas em sua arquitetura que permitem mensagens sem autenticação.

f) **Técnicas de reconhecimento de padrões**

De acordo com (BISHOP, 2006), reconhecimento de padrões são técnicas utilizadas na descoberta de padrões existentes em dados através da utilização de algoritmos computacionais que possibilitam a classificação dos dados em categorias. Esse tipo de técnica é muito utilizado em classificação de spam. Algumas abordagens utilizadas são:

- Bayesianas;
- Redes Neurais Artificiais;
- Sistemas de regras fuzzy;
- Arvores de Decisão; e
- SVM (Support Vector Machines)

As abordagens que classificam *emails* baseados em reconhecimento de padrões podem ser divididas em dois tipos principais: (*i*) As técnicas que utilizam primordialmente a frequência da ocorrência de palavras na mensagem; (*ii*) As técnicas que verificam diversas características

presentes no *email*, tanto no cabeçalho quanto no corpo. As últimas tentam detectar a utilização de recursos da linguagem $HTML$ na mensagem.

2.4.1 Classificadores Bayesianos

A teoria bayesiana de decisão é uma abordagem estatística sistemática para a tomada de decisões fundamental para estudos envolvendo classificação de padrões (DUDA; HART; STORK, 2012). O classificador *Naive Nayes* (NB) é um dos mais usados na classificação de *email* (SCHNEIDER, 2003). Por ser um classificador simples e fácil de implementar, com boa taxa de acerto em comparação com outros algoritmos baseados em aprendizagem de máquina, o NB é muito utilizado por sistemas comerciais de filtragem de *spam* (CHEN; TIAN; ZHANG, 2008). Um estudo comparou três propostas que utilizam o NB (FRANK; HALL; PFAHRINGER, 2002)), *Locally Weighted Naive Bayes* (LWNB), *Discretized Locally Weighted Naive Bayes* (LWNBD) e *Lazy Bayesian Rules* (LBR), com outras propostas utilizando, *Averaged One-Dependence Estimators* (AODE), *K-nearest Neighbours With Distance Weighting* (KNNDW) . O estudo propôs melhorias no algoritmo NB com o objetivo de provar que a utilização do NB melhorado é, em muitos casos, a melhor escolha comparada a outros métodos tradicionais, incluindo o método NB tradicional. Neste estudo foram utilizados 37 *datasets* e os algoritmos modificados apresentaram melhores resultados de classificação.

2.4.2 Frequência de Palavras

O trabalho desenvolvido por (DRUCKER; WU; VAPNIK, 1999) apresentou dois conceitos muito utilizados em técnicas de detecção por frequência de ocorrências de caracteres sendo eles: TF (*Term Frequency*) - Quantidade de vezes que uma palavra aparece em uma mensagem; e TF-IDF (*Term Frequency - Inverse Document Frequency*). O trabalho ainda considera a importância de um termo dentro de uma coleção de documentos e ainda, uma lista de termos que não devem aparecer dentro do vetor de características, *stop list*. O trabalho ainda apresenta algumas técnicas para melhoramento da performance do classificador e para a validação dos resultados. (KIRITCHENKO; MATWIN, 2011) utilizaram uma técnica chamada *co-training* que permite o desenvolvimento de um classificador com um pequeno número de amostras rotuladas, o método sugere que um classificador fraco, construído com amostras rotuladas, pode encontrar amostras similares em uma base ainda não rotulada. Dessa forma, aos poucos, vão sendo inseridas novas amostras rotuladas com a intenção de alimentar a base e refazer o treinamento. Neste

experimento, foram realizados 50 treinamentos na base e, na medida em que mais mensagens eram classificadas, a taxa de acerto acrescia. Na primeira parte do treinamento, o classificador conseguiu atingir 90% de acerto, contudo, após 50 iterações essa taxa aumentou para 94%.

2.4.3 Arvore de decisão

Os autores (BRAGA; LADEIRA, 2007), demonstraram uma abordagem que leva em consideração a característica dinâmica de um *spam*, ou seja, o grande número de variações de técnicas utilizadas em uma mensagem com o objetivo de burlar os sistemas de classificação. Nesta abordagem é proposto três módulos para pré-processamento, classificação e adaptação. Na etapa do pré-processamento, a mensagem é transformada em números utilizando o método de Huffman adaptativos (árvores FGK) e conta com um algoritmo para ordenar as mensagens dentro de vetor. Um dos benefícios da utilização de árvores adaptativas, é a possibilidade de acrescentar novas folhas sem a necessidade de se criar uma nova árvore. Na classificação foi utilizado um classificador SVM. Na fase de adaptação, foi utilizada uma técnica chamada envelhecimento exponencial, que retreina o classificador usando as novas mensagens que chegam. Somente as mensagens mais recentes são consideradas.

A abordagem considera que o spam pode mudar ao longo do tempo e, para isso, dá maior importância as mensagens mais recentes. Ao utilizar o modelo de Arvore de *Huffman* Adaptativa - FGK, o processo de treinamento é acelerado comparado a outras técnicas que utilizam texto puro.

Para (KIRAN et al., 2009), técnicas de reconhecimento de padrão estão entre as abordagens mais utilizadas para classificação de spam. Vários trabalhos procuram comparar técnicas para analisar qual a mais eficiente (GUZELLA; CAMINHAS, 2009). Contudo, tais técnicas estão ao alcance dos *spammers*, que podem explorar vulnerabilidades almejando um modo de burlar o classificador.

Embora ainda não exista uma técnica capaz de classificar corretamente 100% das mensagens, as técnicas de reconhecimento de padrão se mostram eficazes conseguindo bons resultados comparado às demais técnicas comentadas nesta seção. Porém, devido à grande multiplicidade de técnicas de envio de spam, as mensagens terão sempre detalhes específicos que dificultarão a sua correta classificação, além disso, se os sistemas não evoluírem de acordo com as técnicas de envio, ficarão rapidamente obsoletos, uma vez que as técnicas empregadas na prática de

spamming (envio de *spam*), sofrem constantes alterações de acordo com a eficiência dos filtros antispam (OLIVO; SANTIN; OLIVEIRA, 2010).

2.4.4 Outras Técnicas de Reconhecimento de *spam*

Diante da realidade em que não existe atualmente uma técnica capaz de detectar corretamente todas as mensagens de *emails* oriundas de todas as técnicas de disseminação existentes, o *spam* é um problema que ainda não foi solucionado.

(OLIVO; SANTIN; OLIVEIRA, 2010) diz que abordagens para classificação de *spam* utilizam listas de remetentes como parâmetro, os *emails* autênticos oriundos das listas brancas ou *whitelists* e os *spams* originários de listas negras ou *blacklists*, esta ultima, bloqueia o recebimento de mensagens de servidores com o endereço de IP listados em sua base.

Esse tipo de abordagem causa grandes problemas quando o remetente utiliza o servidor SMTP de algum provedor (e.g *Hotmail*, *Gmail*,etc), pois bloqueia todos os outros remetentes que utilizam tais serviços. Não é eficiente bloquear somente o remetente, uma vez que o *email* utilizado para envio da mensagem pode ter sido inventado, sequestrado ou até mesmo furtado de um usuário legítimo.

Para (DAMIANI et al., 2004), outras técnicas contam com a cooperação do usuário para auxiliar o classificador de mensagens. Porém, (DIMMOCK; MADDISON, 2004) concorda que tal enfoque pode não ser totalmente eficiente visto que o usuário não é um especialista capaz de classificar corretamente uma mensagem de *spam*, o que acarretaria na classificação de uma mensagem como *spam*, pelo fato de o usuário simplesmente não gostar de receber o tipo de conteúdo, por uma questão pessoal, complicando ainda, pela subjetividade de cada indivíduo.

Para (LIU; STAMM, 2007) e (SORANAMAGESWARI; MEENA, 2011), as técnicas que se baseiam em aprendizagem de máquina, frequentemente acabam falhando quando um novo tipo de mensagem de *spam* é recebido, uma vez que é necessário treinar um novo modelo, capaz de detectar um novo tipo de mensagem, o *spam* pode ter atingido vários usuários.

Concomitantemente alguns autores acreditam que, a utilização de imagens nas mensagens, como comentado anteriormente, passou ser uma das técnicas mais exploradas para bloquear os mecanismos de antispam (XU et al., 2011), (GAO et al., 2008), (BIGGIO et al., 2007).

Outro problema, encontrado na utilização de classificadores baseados na ocorrência de palavras é a técnica de ofuscamento de palavras ou caracteres. Nesta abordagem quantidade ou ordem dos caracteres contidos nas palavras são trocados, com o objetivo de confundir os

mecanismos de detecção que se baseiam na ocorrência de palavras comuns em spams, uma vez que basta apenas a troca de um caractere para que a palavra não seja classificada como a mesma *string* de caracteres.

Diante do exposto, fica evidente que os mecanismos antispam precisam se adaptar constantemente, devido às inúmeras possibilidades exploradas pelos *spammers*. Um sistema antispam estático possivelmente estaria fadado ao fracasso, visto que as técnicas de envio e camuflagem evoluem constantemente.

Como o *spam* é um problema que possui características que mudam ao longo do tempo, é importante que os sistemas classificadores tenham a capacidade de evoluírem para não se tornarem obsoletos quando uma ou mais particularidades são atribuídas às mensagens. Dessa forma, os sistemas inteligentes evolutivos, tratados no capítulo seguinte, são abordagens apropriadas. Elas podem apresentar bons resultados e, ainda, se adaptar sempre que uma nova técnica começar a ser utilizada nas mensagens de *spam*.

3 SISTEMAS INTELIGENTES EVOLUTIVOS

Neste capítulo são apresentados os principais conceitos sobre os sistemas evolutivos inteligentes utilizados nesta pesquisa. Os modelos apresentados foram selecionados baseados nas recentes pesquisas em fluxos de dados *online*. Por hipótese estes modelos podem superar os modelos classificadores não-adaptativos atualmente empregados ao problema de classificação de *spam online*.

3.1 Introdução

Nos dias atuais, os sistemas de informação são utilizados em quase todas as áreas de atuação humana. De pequenos dispositivos residenciais à grandes sistemas utilizados em empresas, os sistemas de informação se tornaram imprescindíveis para a realização e suporte à diversas tarefas com maior eficiência. Em muitos casos, os sistemas de informação podem substituir a ação humana na condução ou realização de tarefas. O objetivo é proporcionar segurança, conforto, rapidez, gerar lucro, etc, criando assim, ambientes automatizados. As relações humanas bem como os processos que ocorrem na natureza são demasiadamente complexas. Os problemas encontrados nestas relações necessitam de sistemas que possuem características "inteligentes", tais como, aprendizado, raciocínio baseado em conhecimento e adaptação.

Com o objetivo de desenvolver sistemas de informação inteligentes, várias abordagens tem sido propostas na literatura, tais como, redes neurais artificiais, sistemas fuzzy, algoritmos evolucionários. Entretanto, mais recentemente, pesquisadores da área da inteligência computacional observaram que para problemas complexos, que mudam de modo online, é necessário uma abordagem diferente, relativamente pouco explorada, a adaptabilidade online (LUGHOFER; ANGELOV, 2011). A adaptabilidade online é uma característica que capacita os sistemas inteligentes a resolver problemas online em ambientes dinâmicos e não estacionários (LUGHOFER, 2011). Esse tipo de ambiente é tipicamente encontrado em áreas como: medicina, indústria, aeroespacial, automotiva, militar, redes sociais, dentre outras.

Um novo campo de pesquisa surgiu oriundo da área de inteligência computacional. Ele se dedica a pesquisa dos chamados sistemas inteligentes evolutivos (KASABOV, 2007), (ANGELOV, 2009).

Em concordância com (KASABOV, 2007), os sistemas evolutivos são sistemas inteligentes, em geral baseados em redes neurais artificiais, sistemas de regras fuzzy ou redes neuro-fuzzy. Eles são capazes de, a partir de dados de entrada provenientes de ambientes online,

determinar gradualmente suas estruturas e parâmetros (LEITE; COSTA; GOMIDE, 2009). Vários trabalhos na literatura apresentam sucesso na utilização destes sistemas para aplicações que envolvem modelagem, controle, classificação e previsão (ANGELOV, 2009), (ŠKRJANC, 2015).

Os sistemas inteligentes evolutivos são diferentes daqueles denominados adaptativos. Os últimos têm como objetivo ajustar apenas seus parâmetros frente ao problema abordado (ÅSTRÖM; WITTENMARK, 2013). Sistemas evolutivos são diferentes dos sistemas evolucionários. Estes empregam os algoritmos evolucionários baseados em operadores de seleção, recombinação e mutação, inspirados na genética (ABRAHAM; GROSAN; PEDRYCZ, 2008).

Em concordância com (MACIEL et al., 2012), sistemas inteligentes evolutivos são processos que se desenvolvem de modo contínuo e adaptam interativamente sua estrutura e conhecimento oriundos de sequências de dados, aprimorando seu desempenho. (ANGELOV, 2009) define ainda que, sistemas inteligentes evolutivos não possuem estruturas fixas e estas não são definidas *a priori*, ou seja, ela se desenvolve naturalmente à medida em que o fenômeno ou processo físico ou virtual também evolui. Seus parâmetros são adaptados durante tal evolução e ainda, o seu funcionamento é contínuo, o aprendizado se da de modo online.

Os trabalhos iniciais publicados, encontrados na literatura, são da década de 1990 e se baseavam nas redes neurais artificiais (FRITZKE, 1994) e também em redes neurofuzzy (KASABOV, 1996) para a criação de sistemas inteligentes evolutivos. As estruturas dessas redes são bastante flexíveis e, por isso, podem ser alteradas durante o processo evolutivo. Inicialmente, duas abordagens para construção da rede foram sugeridas: os nós (ou nodos) da rede já existem antes do treinamento e somente as conexões entre os nós são criadas durante o treinamento, ou tanto os nós quanto as conexões são criadas durante o treinamento. Esses modelos iniciais foram aprimorados em trabalhos consecutivos, e assim, deram origem a outros modelos baseados em redes neurais artificiais evolutivas (KASABOV, 2001), (KASABOV; SONG, 2002).

Em meados dos anos 2000, outros trabalhos publicados propuseram sistemas inteligentes evolutivos que se baseavam em modelos de regras flexíveis (ANGELOV, 2013). Nestes modelos, um conjunto de regras flexiveis fuzzy realizam o mapeamento não linear entre entradas e saídas, utilizando os modelos fuzzy tipo Mamdani ou tipo Takagi-Sugeno. Um algoritmo de aprendizado on-line faz a identificação dos parâmetros. Tanto a estrutura do modelo (base de regras) quanto os seus parâmetros são modificados durante o processo evolutivo (LEITE et al., 2012).

Angelov e Kasabov (ANGELOV et al., 2006), concordam que os métodos utilizados em sistemas inteligentes evolutivos podem ser divididos em duas categorias principais de aprendizado; a aprendizagem direta e aprendizagem indireta. A aprendizagem direta utiliza métodos de aprendizagem supervisionada e propõe uma solução não-linear ao processo que por sua vez é resolvido numericamente. A aprendizagem indireta é realizada por métodos que particionam os dados em grupos de forma não supervisionada, e utilizam métodos de aprendizagem supervisionados, como por exemplo, o método de mínimos quadrados recursivos.

Um processo evolutivo pode se manifestar de diferentes modos (KASABOV, 2007) : i) aleatório: Quando não há regras que definem o processo, sendo este imprevisível; ii) caótico: Quando o processo é previsível apenas no curto prazo, relacionando o momento atual com o momento anterior através de uma função não-linear; iii) quase-periódico: O processo é previsível, porém está sujeito a erros. As mesmas regras se aplicam ao longo do tempo, mas com pequenas mudanças a cada iteração; iv) periódico: O processo repete os padrões ao longo do tempo, sendo totalmente previsível, ou seja, possuindo regras claras que regem este processo.

O processo de envio de *spam*, conforme tratado neste trabalho, utiliza-se de diversas técnicas de envio, baseadas no conteúdo da mensagem, forma de envio ou mesclando as duas técnicas, a fim de conseguir atingir o máximo de usuários, burlando os sistemas classificadores.

Dado que, por não existirem regras que definem com precisão os processos envolvidos para envio de mensagens de *spam* e, ainda, por não ser possível prever o tipo de técnica utilizada no envio de *spam* pelos *spammers*. O processo tratado neste trabalho se apresenta de forma aleatória devido ao fato de que os *spams* provem de diversas fontes que se utilizam de vários tipos de técnicas, tratadas no capítulo 2 deste trabalho, sendo impossível prever quais técnicas serão utilizadas tornando o fluxo de dados aleatório. Caótico, pois em um curto prazo, o fluxo de dados de *spam* pode ser previsível, em determinadas épocas ou eventos. Contudo, este fluxo pode mudar completamente devido ao surgimento de novas técnicas de spam ou mudanças nos protocolos utilizados para envio e recebimento da mensagem.

3.2 Sistemas Fuzzy Evolutivos

Sistemas fuzzy evolutivos podem ser vistos como a combinação entre: sistemas fuzzy, que são utilizados como um mecanismo de compactação e representação da informação de modo evolutivo; e métodos recursivos de aprendizado de máquina (KASABOV; FILEV, 2006).

Esses sistemas, em sua maioria, utilizam informação sobre a organização espacial das variáveis de entrada, ou entrada e saída, para definir um conjunto de regras de forma adaptativa, a partir de um fluxo de dados. Em geral, a organização espacial dessas variáveis é estimada através de algoritmos de agrupamento recursivos não supervisionados (OLIVEIRA; PEDRYCZ, 2007), esses algoritmos são capazes de processar novas amostras de forma incremental, adaptando os grupos existentes. Os grupos, por sua vez, representam uma decomposição evolutiva do espaço de entrada (ou entrada e saída), gerando regras fuzzy, que podem ser adicionadas ou excluídas de um modelo com estrutura flexível, baseada na mudança dinâmica dos dados. Os parâmetros restantes do modelo são ajustados por métodos recursivos de aprendizado de máquina. Esses sistemas têm como característica principal, a capacidade de adaptar seus parâmetros e, sobretudo, sua estrutura de forma independente em modo online, à medida que variações nos dados de entrada são detectadas. Esses sistemas representam uma mistura fuzzy de modelos (geralmente lineares) locais simples, cuja combinação não é linear e adaptativa, sendo individualmente apropriados para processamento de dados não estacionários em de modo online, ou processamento de uma grande massa de dados, em que a utilização de métodos tradicionais acarreta em um elevado custo computacional (OLIVEIRA; PEDRYCZ, 2007).

De acordo com (ANGELOV; ZHOU, 2008) o termo “evolutivo” é utilizado para diferenciar tais sistemas de sistemas adaptativos, pois esses sistemas, além de serem capazes de ajustar seus parâmetro a partir de dados (o que geralmente é referido atribuído ao termo adaptativo) também possuem a característica de adaptarem sua estrutura. Dessa maneira, o termo “evolutivo” foi utilizado para definir um nível mais alto de adaptação, em que, diferentemente de sistemas adaptativos, a estrutura do sistema não é fixa.

Além disso, como já foi colocado, o termo “evolutivo” não deve ser confundido com o termo “evolucionário”. Algoritmos evolucionários, tais como algoritmos genéticos (GOLDBERG; DEB; KORB, 1989) e programação genética (KOZA, 1992), baseiam-se no processo de evolução que ocorre em populações de indivíduos e utilizam operadores baseados nos conceitos de seleção, cruzamento e mutação como mecanismos de adaptação. Já os sistemas fuzzy evolutivos baseiam-se no processo de evolução de indivíduos ao longo de suas vidas, principalmente nos processos humanos de aprendizagem, baseado na geração e adaptação de conhecimento a partir de experiências (ANGELOV; ZHOU, 2008). Assim, uma definição mais simples para o mecanismo de aprendizagem desses sistemas seria o processo de aprendizagem de um indivíduo ao longo de sua vida. Esse processo tem início a partir de um conjunto vazio de regras (conhe-

cimento) e novas regras são aprendidas à medida que o indivíduo vivencia novas experiências, essas experiências não podem ser explicadas pelas regras existentes.

O processo de geração de regras é gradual e as regras não são fixas ou pré-determinadas. Além disso, regras podem ser revisadas para melhor se adaptarem às experiências vivenciadas. O interesse por essa nova abordagem para o problema de modelagem fuzzy vem crescendo nos últimos anos e diversas pesquisas tem apontado como uma abordagem eficiente na solução de diversos problemas.

3.3 Modelagem Evolutiva Granular Fuzzy

Modelagem evolutiva granular fuzzy (FBeM), é um método de computação granular, pois utiliza informação do tipo fuzzy para construir mapas granulares que associam os espaços de entrada e saída levando em conta que os dados podem ser valores numéricos ou valores incertos. Grânulos fuzzy generalizam a ideia de processamento de dados numéricos para processamento de dados fuzzy. Modelagem granular fuzzy oferece algoritmos e modelos com regras simples descrevendo o significado do mapeamento granular entre espaços (LEITE; COSTA; GOMIDE, 2009). A utilização de conjunto fuzzy para dados incertos é a estratégia utilizada por FBeM para lidar distúrbios, julgamento especialista, opiniões e imprecisões (LEITE et al., 2015).

FBeM se beneficia objetos fuzzy granulares ao sumarizar a informação que chega auxiliar na tomada de decisão. Modelos no FBeM são criados e evoluem conforme os dados chegam através do fluxo de dados. Fontes de dados suportados pelo FBeM incluem, tráfego de dados *Web*, dados de áudio e vídeo, dados climáticos, etc. O aprendizado no algoritimo FBeM cria grânulos recursivamente, conforme o fluxo. Eventualmente, o quociente da estrutura granular pode ser otimizado, redefinido, deletado e fundido conforme a necessidade (LEITE et al., 2015).

Grânulos consistem de conjuntos de objetos com características semelhantes como equivalência, proximidade, funcionalidade ou indistinguibilidade (PEDRYCZ; SKOWRON; KREINOVICH, 2008).

O modelo FBeM para classificação é formado por um conjunto de regras fuzzy. FBeM é útil quando para fluxos de dados e algoritmos convencionais apresentam problema de escabilidade. O método FBeM não necessita de conhecimento *a priori* sobre a base de dados para iniciar sua fase de aprendizagem. Portanto, estes modelos aprendem do zero. As regras e grâ-

nulos FBeM são criados dinamicamente e adaptadas ao longo do tempo conforme necessidade (LEITE et al., 2012).

Para cada grânulo de informação FBeM existe uma regra correspondente a este. Segundo (LEITE et al., 2015), a parte antecedente das regras FBeM é composta por hiper retângulos fuzzy, e a parte consequente é composta de termos funcionais e linguísticos. Uma regra FBeM é como segue:

$$\begin{aligned} R^i : \text{Se}(x_1 \text{ é } A_1^i) \text{ e } ... \text{ e } (x_j \text{ é } A_j^i) \text{ e } ... \text{ e } (x_n \text{ é} A_n^i) \\ \text{Então } (y_1 \text{ é } B_1^i) \text{ e } \bar{y}_1 = p_k^1(x_j \forall_j) \text{ e } ... \\ \text{e } (y_m \text{ é } B_m^i) \text{ e } \bar{y}_m = p_m^i(x_j \forall_j) \end{aligned} \tag{3.1}$$

onde $(y_m \text{ é } B_m^i)$ é a parte liguística da equação e $\bar{y}_m = p_m^i(x_j \forall_j)$ é a parte funcional. x_j e y_j são variáveis provenientes do fluxo de dados $(x,y)^{[h]}$, onde $h = 1,2,...$. A_j^i e B_k^i são funções de pertinência trapezoidais criadas conforme os dados; p_k^i representa uma função afim.

O consequente de uma regra FBeM é composto por uma T-norma entre um conjunto fuzzy e uma função afim. Estes fornecem uma aproximação da saída e limites de tolerância na aproximação.

Cada regra FBeM ao menos parcialmente ativada por uma amostra de dados contribui para a composição da saída do sistema. A saída do sistema FBeM para classificação é dada pelo valor da média ponderada de todas as regras criadas, conforme:

$$p_k = \frac{\sum_{i=1}^{c} min(A_1^i, ..., A_n^i) p_k^i}{\sum_{i=1}^{c} min(A_1^i, ..., A_n^i)} \tag{3.2}$$

No caso de problemas de classificação, o polinômio p é de ordem 0, e assume valores inteiros.

A máxima largura que um conjunto fuzzy A_j^i pode assumir em FBeM é limitada por ρ. Onde,

$$largura(A_i^j) = L_j^i - l_j^i \leq \rho, \tag{3.3}$$

para $j = 1,...,n$; $i = 1,...,c$. Assumindo diferentes valores para ρ, diferentes representações dos fluxos de dados em diferentes níveis de granularidade são apresentadas. Seja a região de expansão de um conjunto A_j^i ser denotada por:

$$E_j^i = [mp(A_j^i) - \frac{\rho}{2}, mp(A_j^i) + \frac{\rho}{2}] \tag{3.4}$$

onde $mp(A_j^i)$ é ponto central de A_j^i. A região de expansão auxilia o FBeM a determinar quando uma amostra deve ser considerada ou não em um grânulo já existente.

Para dados normalizados o valor de ρ está entre $[0,1]$. Se o valor de ρ é igual à 0, o grânulo é incapaz de expandir. Portanto, apenas uma regra FBeM é criada para cada amostra, o que pode não ser uma representação adequada pelo elevado nível de detalhamento. Se o parâmetro ρ tem valor 1, o método FBeM cria um único grânulo, ou seja, uma única regra para representar todo o espaço de dados, o que pode não ser ideal em um ambiente não-estacionários e *online*.

O FBeM utiliza um procedimento simples para adaptar a granularidade ρ continuamente, como segue. Sendo r a diferença entre o número de grânulos corrente e o número de grânulos h_r passos anteriores, $r = c^{[h]} \breve{} c^{[h-h_r]}$. Se a quantidade de grânulos aumentar mais do que uma dada taxa de crescimento η, ou seja, $r > \eta$, então ρ é incrementado:

$$\rho(novo) = (1 + \frac{r}{h_r})\rho(velho) \tag{3.5}$$

A equação 3.5 controla o valor de ρ de modo que rejeita grandes valores e, portanto, evita o aumento de complexidade. Se o valor de ρ cresce à uma taxa menor do que o valor η, ou seja, $r \leq \eta$, então o valor de ρ decresce conforme:

$$\rho(novo) = (1 - \frac{\eta - r}{h_r})\rho(velho) \tag{3.6}$$

Este procedimento mantém o modelo de granularidade FBeM variando dinamicamente conforme o fluxo de dados *online*. Não necessariamente existem regras FBeM antes do início do procedimento de aprendizagem. Logo, o procedimento incremental para criação de regras ocorre sempre que uma entrada $(x_1,...,x_n)$ não pertence à região de expansão $(E_1^i,...,E_n^i)$, $i = 1,...,c$, ou a saída y não pertence à expansão E_k^i, $i = 1,...,c$. Caso contrário, a base de regras segue inalterada e os parâmetros das funções de pertinências trapezoidais dos termos antecedentes de regras são adaptados.

O algoritmo de aprendizagem FBeM é como segue.

Figura 3.1 – Algoritmo de aprendizagem: FBeM

Algoritmo de aprendizagem: FBeM

DEFINIR os parâmetros p,h_r,η,ψ,c;
LER $(x,y)^{[h]}$, $h = 1$
CRIAR grânulo y^{c+1};
  PARA $h = 2,...$ **FAZER**
    LER $(x,y)^{[h]}$
    FORNECER aproximações singulares $p(x^{[h]})$;
    CALCULAR o erro $\varepsilon^{[h]} = y^{[h]} - p(x^{[h]})$;
    SE $y^{[h]}$ ou $x^{[h]}$ não estão na região de expansão dos grânulos $E^i \forall i$;
      CRIAR grânulo γ^{c+1};
    SENÃO
      ADAPTAR o grânulo mais ativo γ^i,$i = \max_i(S(x,A^1),...,(S(x,A^c))$;
    FIM SE
  SE $h = \alpha$, $\alpha = 1,2...$
    COMBINAR os grânulos semelhantes
    ATUALIZAR a granularidade ρ
    REMOVER os grânulos inativos
  FIM SE
FIM

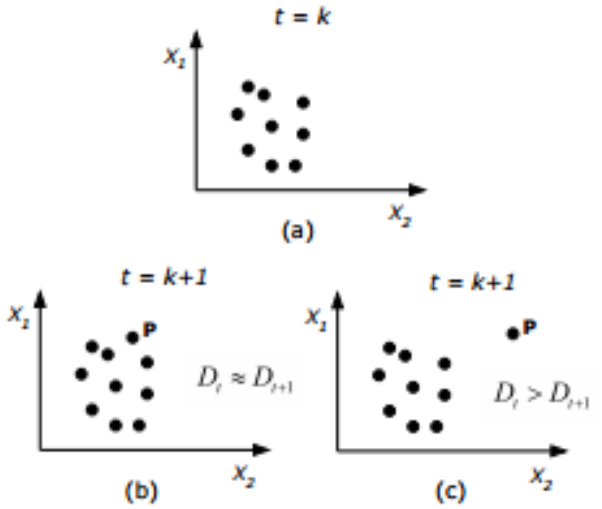
Fonte: Adaptado de (LEITE et al., 2015)

3.4 Estimação Recursiva da Densidade dos Dados

O *Recursive Density Estimation* (RDE) é uma técnica para detecção de *outliers* em um conjunto de dados n-dimensional proposto por (COSTA; ANGELOV; GUEDES, 2015). O método é baseado no cálculo da densidade do conjunto de dados analisado que é indicativo da proximidade desses dados uns dos outros no instante de tempo. Ao analisar esta densidade, podemos estimar se uma amostra de dados se encontra fora da distribuição normal das demais, configurando um *outlier*.

Por exemplo, considerando a Figura 3.2 onde, em um conjunto de dados bidimensional, os dados da Figura 3.2 (a) estão distribuídos e possuem densidade D_t em um instante de tempo k. Se em um instante seguinte $k = 1$ surge um novo ponto P e este, esteja próximo dos outros pontos, conforme Figura 3.2 (b) o valor da densidade no instante 1 , D_{t+1}, será próximo ao valor da densidade do instante anterior visto que P está perto dos demais pontos.

Figura 3.2 – Conjunto de dados bidimensional



Fonte: Adaptado de (KOLEV et al., 2013)

Na Figura 3.2(c), o ponto P está longe dos demais pontos. Isso faz com que a o valor da densidade D_{t+1} aumente em relação a D_t.

Dessa forma, quando um possível *outlier* surge ao *RDE*, o valor da densidade do conjunto de dados tenderá a cair em relação ao instante anterior D_{t-1}, essa diferença tende a aumentar na medida em que o ponto estiver mais distante dos demais pontos do conjunto de dados.

Se considerarmos um cluster C, formado por um conjunto de amostras de dados, onde cada uma dessas amostras é um vetor $x \in R^n$, a função de densidade local desse cluster d_C, em relação a amostra x no instante k, é calculada usando a seguinte equação (KOLEV et al., 2013):

$$d_C(x_k) = \frac{1}{1 + \frac{1}{n_C}\sum_{i=1}^{N_C} \|x_k - x_i\|^2}. \tag{3.7}$$

onde N_C é a quantidade de amostras de C, nesta equação, a distância euclidiana é utilizada para medir a distância entre x_k e as demais amostras pertencentes a C.

Se em uma aplicação online, onde uma nova amostra é coletada em intervalos regulares em cada instante de tempo k, o cálculo intervalar da densidade desse conjunto de dados, pode se tornar inviável ao utilizar a equação acima. Para utilizar a equação proposta para calcular a densidade no instante de tempo k, seria preciso armazenar todas as amostras anteriores ao instante k, o que exigiria maior custo computacional e quantidade de memória de armazenamento.

Para solucionar tal problema, (KOLEV et al., 2013) propôs a seguinte expressão par ao cálculo da densidade das amostras:

$$D(x_k) = \frac{1}{1 + \|x_k - u_k\|^2 + x_k - \|\mu^2\|}. \tag{3.8}$$

A média de todas as amostras de dados é representada por u_k e o produto escalar médio dessas amostras é representado por x_k. Dessa forma, o cálculo da densidade de um conjunto de dados em um instante k pode ser obtido armazenando somente valores de μ e x no instante $k-1$ proporcionando maior agilidade no algoritmo com menor custo computacional e uso de memória. Outra vantagem do RDE é que o mesmo não necessita da estimação de nenhum valor de parâmetro ou treinamento prévio nos seus cálculos.

3.5 Tipicidade e Excentricidade

TEDA (*Typicality and Eccentricity Data analytics*) é um método proposto por (ANGELOV, 2014) que se baseia nos conceitos de excentricidade e tipicidade que representam a densidade e proximidade de dados da estimativa da densidade recursiva (RDE) apresentado na seção anterior, porém com melhorias em sua formulação. O TEDA tem por objetivo generalizar e evitar suposições restritivas, herdadas de métodos estatísticos tradicionais e teoria da probabilidade, tais como independência de amostras de dados, incapacidade de trabalhar com grandes conjuntos de dados e outras suposições de métodos de distribuição de dados (e.g gaussianos).

Abordagens estatísticas tradicionais são frequentemente utilizadas em processos aleatórios, porém podem ignorar a dependência dos dados em processos reais como clima, economia, biologia, sociologia e psicologia. Tais dados, na maioria das vezes, são complexos, incertos e pouco conhecidos, porém, não são em sua totalmente aleatórios. O TEDA é uma metodologia sistemática que não necessita de dados anteriores e pode ser utilizado para vários métodos de processamento de imagens, clusterização, classificação, previsão, controle, filtragem, regressão etc, (ANGELOV, 2014).

Dessa forma, o TEDA se apresenta como uma abordagem estatística alternativa, capaz de trabalhar com qualquer tipo de processos com dados aleatórios, onde cada observação é completamente independente da outra (SOARES et al., 2017).

Sendo $X = [x_{(1)}, ..., x_{(K)}]'$, $X \in \Re^{K \times n}$, um conjunto de dados; $k = 1, ..., K$ um índice de tempo. A soma das distâncias, $\pi(.)$, da k-ésima amostra, $x_{(k)}$, para todas as amostras de X é dada por:

$$\pi(x_{(k)}) = \sum_{j=1}^{K} d(x_{(k)}, x_{(j)}), \tag{3.9}$$

onde $d(.)$ pode ser qualquer métrica de distância. Recursivamente, $\pi(.)$ no índice de tempo k, i.e., a soma das distâncias entre as k-ésimas amostras de dados e a amostra anterior de um fluxo de dados é dada por:

$$\pi(x_{(k)}) = \pi(x_{(k-1)}) + d(x_{(k)}, x_{(k-1)}). \tag{3.10}$$

A excentricidade $\varepsilon(.)$ de $x_{(k)}$ em relação a todas as amostras contidas na base de dados é dada por :

$$\varepsilon(x_{(k)}) = \frac{2\pi(x_{(k)})}{\sum_{j=1}^{K} \pi(x_{(j)})} \tag{3.11}$$

a tipicidade $\gamma(.)$ é o complemento da excentricidade, i.e.,

$$\gamma(x_{(k)}) = 1 - \varepsilon(x_{(k)}). \tag{3.12}$$

Assumindo $d(.)$ como norma-2, a excentricidade pode ser calculada por:

$$\varepsilon(x_{(k)}) = \frac{1}{k} + \frac{(\mu_x^k - x_{(k)})^T (\mu_x^k - x_{(k)})}{k[\sigma_x^k]^2} \tag{3.13}$$

onde a tipicidade pode ser calculada recursivamente através de :

$$\gamma(x_{(k)}) = \frac{k-1}{k} - \frac{(\mu_x^k - x_{(k)})^T (\mu_x^k - x_{(k)})}{k[\sigma_x^k]^2}. \tag{3.14}$$

A média μ_x^k, média dos quadrados $\mu_{x^T x}^k$, e variância $[\sigma_x^k]^2$ das nuvens de dados são atualizadas através de:

$$\mu_x^k = \frac{(k-1)\mu_x^{k-1}}{k} + \frac{x_{(k)}}{k}, \tag{3.15}$$

$$\mu_{x^T x}^k = \frac{(k-1)\mu_{x^T x}^{k-1}}{k} + \frac{x_{(k)}^T x_{(k)}}{k}, \tag{3.16}$$

e

$$[\sigma_x^k]^2 = \mu_{x^T x}^k - [\mu_x^k]^T \mu_x, \tag{3.17}$$

sendo $\mu_x^0 = \mu_{x^T x}^0 = [\sigma_x^0]^2 = 0$.

3.6 TEDA Class

O algoritmo TEDA para classificação pode ser entendido como uma extensão do *framework* TEDA, porém este requer informações adicionais sobre as definições das classes de cada regra fuzzy criada.Para isto, portanto, suas regras são alteradas (KANGIN; ANGELOV, 2015a). A regra fuzzy no TEDA Class é dada por:

$$R_i(x) : \text{Se } (x \sim x_i^*) \text{ então } y_i \tag{3.18}$$

onde $i = 1,...,N$; Θ_i é uma matriz de coeficientes, e $\sim$ indica o relacionamento de uma amostra x à regra R_i; e x_i^* é um ponto representativo na nuvem de dados (ponto focal). Em particular, a regra R_i mapeia as amostras $x \in \Re^n$ que são próximas a i-ésima nuvem a uma classe de saída $y \in \Re$. Este valor de saída está associado à uma classe C, que é finita.

$$R_i(x) : \text{Se } (x \sim x_i^*) \text{ então } y_i \tag{3.19}$$

A excentricidade local, ε_i, e a tipicidade local, γ_i, são definidas conforme as amostras são assimiladas a i-ésima nuvem de dados. Uma nuvem e uma regra correspondente são criadas quando o valor da tipicidade local $\gamma_i \; \forall i$ é menor que um limiar α, i.e.,

$$\max_{i=1,...,N} \gamma_i(x_{(k)}) \leq \alpha. \tag{3.20}$$

caso contrário, a amostra $x_{(k)}$ é atribuída a regra ativa mais relevante. O nível de ativação da i-ésima regra para uma amostra é:

$$w_i(x_{(k)}) = \frac{\gamma_i(x_{(k)})}{\sum_{j=1}^{N} \gamma_j(x_{(k)})} \tag{3.21}$$

Assim que uma amostra é atribuida a uma regra R_{i^*},a excentricidade local, ε_{i^*}, e a tipicidade local, γ_{i^*}, são atualizadas baseadas nas equações 3.13 e 3.14. Além do mais, os parâmetros

consequentes Θ_{i^*} são atualizados através dos Minímos Quadrados Recursivos(MQR) (GRANT, 1987). As estimativas de saída são dadas por:

$$\bar{y}_{(k)} = \sum_{i=1}^{N} w_i \bar{y}_{i(k)}. \tag{3.22}$$

O valor de cada classe se torna disponível após a estimação desta. O algoritmo de apendizagem do TEDA Class é dado abaixo.

Figura 3.3 – Algoritmo de aprendizagem: TEDA Class

Algoritmo de aprendizagem: TEDA Class

DETERMINAR m, $N = 0$;
LER a primeira amostra $x_{(1)}$;
CRIAR a regra R_N, $N = N + 1$;
PARA k = 2, ...
  LER $x_{(k)}$;
  CALCULAR $\gamma_i(x_{(k)})$ e $\varepsilon_i(x_{(k)})$, $i = 1,...,N$;
  CALCULAR os níveis de ativação $w_i(x_{(k)})$, $i = 1,...,N$;
  FORNECER estimação de classe $\bar{y}_{(k)}$;
  SE equação 3.20 for verdade
    CRIAR regra R_{N+1}, $N = N + 1$;
  SENÃO
    ATUALIZAR γ_{i^*} e ε_{i^*} onde $i^* = argmax\ w_i(x_{(k)})$;
    // A saída $y_{(k)}$ se torna disponível
  FIM
FIM

Fonte: Adaptado de (KANGIN; ANGELOV, 2015b)

3.7 Seleção de características utilizando Colônia Artificial de Formigas

Ao longo dos últimos anos, um grande número de dados tem sido gerado por meio dos sistemas de bancos de dados e computadores, com o objetivo de alimentar os sistemas de informação. Normalmente, a informação está ocultada dentro de um conjunto de dados brutos, ou seja, dados sem nenhum tipo de tratamento. Somente após a correta seleção das variáveis a informação poderá ser gerada e, em seguida, poderá ser utilizada como insumo para o conhecimento (KABIR; SHAHJAHAN; MURASE, 2013).

Quanto ao e-mail, suas características, quando selecionadas corretamente, podem despontar informações importantes como, origem, composição, finalidade e tipo de mensagem. No entanto, dado ao fato de que o conteúdo das mensagens podem variar ao longo do tempo e ainda por se tratar de uma tecnologia que, assim como a internet, precisa evoluir, pois novas tecnologias são desenvolvidas a todo o momento, o número de características de uma mensagem de e-mail pode aumentar com o tempo. O SpamAssassin, na versão 3, utiliza uma série de 711 testes com o objetivo de extrair essas características e atribuiu uma pontuação a cada medição(<http://spamassassin.apache.org/old/tests_3_0_x.html>).

No entanto, Informações com grande número de características precisam ser tratadas com o objetivo de reduzir sua dimensionalidade. O objetivo da seleção de características é escolher um subconjunto ótimo de atributos disponíveis, eliminando os dados menos importantes ou desnecessários. Para extrair a maior quantidade possível de informações de um conjunto de dados ao usar um número menor de características, os recursos com pouca ou nenhuma informação preditiva devem ser eliminados e as características redundantes fortemente correlacionadas devem ser ignoradas (KABIR; SHAHJAHAN; MURASE, 2013).

A redução da dimensionalidade da informação usando a seleção de características é uma das etapas mais importantes para o processo de classificação. A seleção de recursos também tem uma importância considerável em áreas como mineração de dados, e reconhecimento de padrões (KANAN; FAEZ; TAHERI, 2007). Além disso, um subconjunto de características mais relevantes ou mais discriminativas facilita a interpretação de modelos, e reduz a chance de sobreajuste e consequentemente pode produzir melhores resultados devido à eliminação de características que podem confundir a descoberta de padrões, tendências e relacionamentos (GUYON; ELISSEEFF, 2003). Do mesmo modo, influencia em outros aspectos da classificação como a precisão do algoritmo, tempo de processamento e custo computacional. Portanto, é necessário reduzir significativamente as informações falsas, isto é, características irrelevantes, redundantes e ruidosas, do conjunto de dados original e, criar um subconjunto de recursos mais acentuados (KANAN; FAEZ; TAHERI, 2007). Em detalhes, a seleção de características é, no entanto, um processo de busca ou técnica que seleciona um subconjunto de recursos importantes para a construção de modelos de aprendizado robustos, como Redes Neurais, Máquina de Vetores de Suporte e Sistemas Fuzzy (KABIR; SHAHJAHAN; MURASE, 2013).

A Computação Evolucionária é um tipo de abordagem que se inspira nos mecanismos evolucionários naturais com o objetivo de desenvolver algoritmos computacionais. Os algorit-

mos desenvolvidos a partir desta abordagem utilizam um conjunto de informações codificadas em um formato que imita a informação genética (cromossomos) e se desenvolve utilizando uma lógica que possibilita que cada unidade possa trocar informações entre si (YAO; LIU, 1996).

Inteligência de Enxames é o termo utilizado para se referir aos sistemas evolucionários que estudam o comportamento coletivo dos indivíduos de uma população, e a busca de tais indivíduos para soluções simples em problemas complexos. Um enxame é a generalização de uma coleção estruturada de indivíduos capazes de interagir uns com os outros. O exemplo clássico de um enxame pode ser o de abelhas. Contudo, é importante salientar que, outros sistemas, como por exemplo, uma colônia de formigas, pode ser definida, também, como um enxame, onde cada formiga é um agente. Outro exemplo de enxame pode ser uma revoada de pássaros, neste caso os agentes são os pássaros. Porém, o termo não está limitado a insetos, um engarrafamento de carros pode ser considerado um enxame onde os carros são os agentes (KABIR; SHAHJAHAN; MURASE, 2013).

Dos estudos relacionados à inteligência de enxames, existem duas grandes linhas de pesquisa. A primeira concentra suas pesquisas no estudo de comportamento de insetos sociais, como abelhas, formigas, vespas, cupins. A segunda pesquisa as habilidades das sociedades humanas e a sua maneira de processar conhecimento. As técnica mais conhecidas de Inteligência de Enxames são a otimização por enxame de partículas, otimização por enxame de formigas, algoritmos de coleta de alimento por bactérias e algoritmos de colônia de abelhas (DORIGO; BIRATTARI; STUTZLE, 2006).

O algoritmo de otimização por colônia de formigas *Ant Colony Optimization* (ACO) é um termo que designa procedimentos gerais de uma classe de procedimentos que se baseiam no comportamento de formigas. A idéia principal da otimização através da colônia de formigas é a reformulação do problema em um grafo, no qual se procura alcançar um caminho ótimo para alcançar determinado objetivo. Formigas artificiais são utilizadas para percorrer este grafo. Formigas artificiais colocam feromônio nas bordas e vértices de um grafo e escolhem o seu caminho em relação a probabilidades que dependem de trilhas de feromônios que foram previamente colocadas pelo colônia (AGHDAM; GHASEM-AGHAEE; BASIRI, 2009).

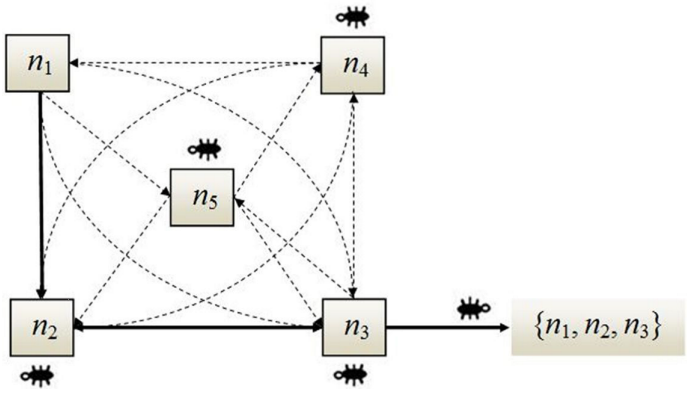
O modelo matemático do comportamento de uma colônia de formigas será descrito a seguir. Dado um grafo G com V vértices, representando um problema, uma formiga artificial é inserida em cada um dos vértices (NEMATI et al., 2009). Cada uma das formigas percorre um caminho de acordo com uma fórmula probabilística baseado na função do feromônio “deixado”

em cada aresta do grafo até chegar ao seu destino.

1. Inicialize os parâmetros , r, a, b e as trilhas de feromônio t_{ij} com o mesmo valor inicial t_0.
2. Coloque cada formiga em uma aresta aleatoriamente selecionada do grafo $G(V,E)$, onde V é o conjunto de vértices e E é o conjunto de arestas de G.
3. Para cada formiga k, em cada aresta até o destino, construa soluções baseadas na regra de transição de estado(P_i^k ,j)
4. Avalie o custo de todas as soluções (calcule f (S)) .
5. Guarde a melhor solução até o momento.
6. Para cada aresta do grafo, aplique a regra de atualização da trilha de feromônio (calcule Δt_{ij}, e t_{ij}).
7. Se condição de término não for alcançada, retorne ao passo 3.

Em ACO para seleção de características, as características selecionadas são representadas através de uma combinação de arcos, nos quais as formigas passaram pelo grafo (NEMATI et al., 2009). Cada formiga na colônia deve visitar cada nó do grafo pelo menos uma vez. Cada formiga inicia em um nó específico, e passa por todos os outros nós e termina o processo em um determinado nó. Cada nó possui dois arcos conectados entre eles, no qual representa seleção ou exclusão da característica a qual esse nó é assimilado. Portanto, a combinação de arcos atravessados juntos oferece uma representação completa de uma solução candidata. A Figura 3.4 ilustra um ACO para seleção de características.

Figura 3.4 – ACO para seleção de características



Fonte: Adaptado de (NEMATI et al., 2009)

O objetivo deste algoritmo de otimização é minimizar o erro de classificação na previsão da saída. Neste abordagem híbrida, o papel de cada formiga é construir um subconjunto de solução. As "formigas"formam soluções aplicando uma decisão probabilística para escolher o próximo nó. Neste caso, cada subconjunto de recurso representa um estado. As regras de

transição do estado ajudam as formigas a selecionar recursos usando a trilha de feromônios e o valor heurístico. Uma formiga escolhe um recurso da seguinte maneira:

$$P_i^k = \{\frac{[\tau_i(t)]^\alpha.[\eta_i]^\beta}{\sum_{u\in j^k}[\tau_u(t)]^\alpha.[\eta_u]^\beta}\} \tag{3.23}$$

onde J^k é o conjunto de recursos viáveis que podem ser adicionados a solução parcial; $\alpha \leq 0$ e $\beta \leq 0$ são dois parâmetros que determinam a importância do valor do feromônio e desejabilidade heurística.

Cada formiga deposita uma quantidade de feromônio que diminui de acordo com sua classificação. Além disso, a melhor formiga até o momento deposita sempre a maior quantidade de feromônio em cada iteração.

A melhor iteração até o momento fornece o maior *feedback*, com peso w ; a melhor formiga da atual iteração contribui para atualização de feromônios com o valor multiplicado pelo peso w e, finalmente, ρ é um parâmetro de quantidade de evaporação.

$$\tau_{ij} \leftarrow \rho * \tau_{ij} + \sum_{r=1}^{w-1}(w-r)\Delta\tau_{ij}^r + w\Delta\tau_{ij}^{bs} \tag{3.24}$$

onde,

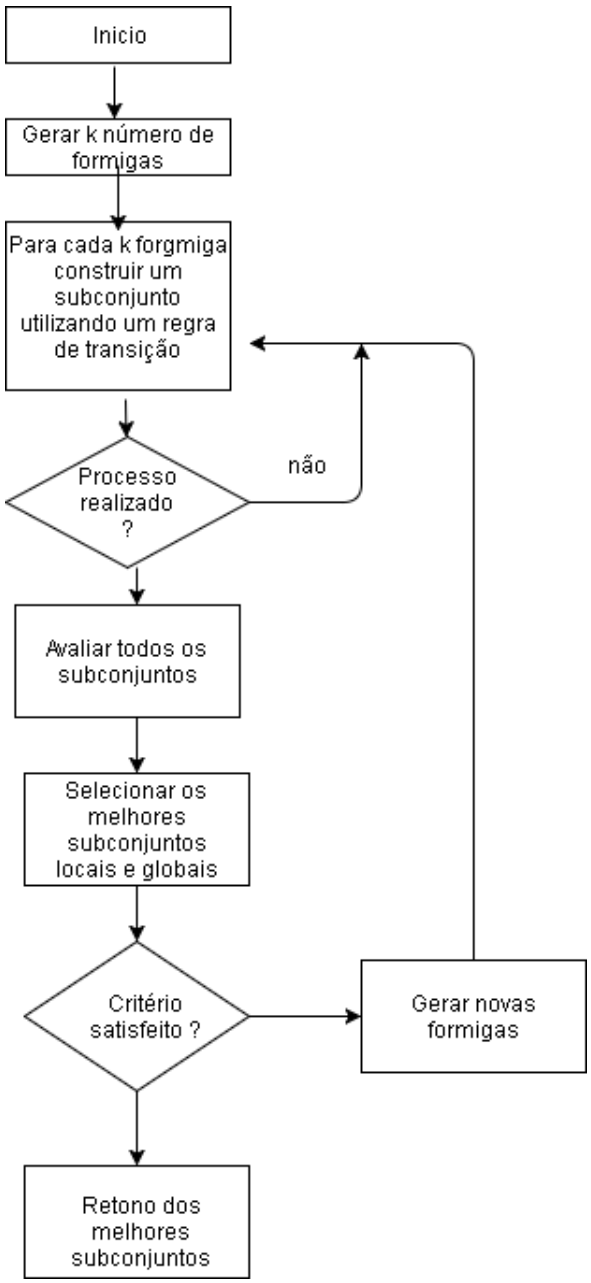
$$\Delta\tau_{ij}(t) = \sum_{k=1}^{n}(\gamma'\frac{(S^k)}{|S^k|}) \tag{3.25}$$

O feromônio é atualizado de acordo com a medida da eficiência do subconjunto de características da formiga γ' e o tamanho do próprio subconjunto.

No caso utilizado neste projeto, os valores de feromônios e heurísticas não são associadas aos ligamentos. No entanto, cada característica tem seu próprio valor de feromônio e heurísticos associados. A base extraída possui 711 características, portanto optou se por reduzir a mesma em outras quatro bases sendo a primeira com as 711 características, a segunda com 533 características correspondendo a 75%, a terceira com 355 caracteristicas cerca de 50% e a quarta com 178 características representando 25% do total das características.

A Figura 3.5 ilustra a essência do esquema para seleção de características realizado pelo método baseado em colônia de formigas. Para maiores detalhes sobre o algoritmo refira-se a (NEMATI et al., 2009).

Figura 3.5 – Conjunto de dados bidimensional



Fonte: Elaborado pelo autor

4 METODOLOGIA

Este capítulo apresenta a metodologia. A primeira seção descreve como a base de de dados foi obtida. Em seguida, o procedimento geral para análise dos modelos classificadores propostos é apresentado.

4.1 Construção da Base de Dados

Como a base de dados de *emails* (*spambase*), amplamente utilizada na literatura não reflete mais a atual situação sobre os *emails* e os *spams* e a forma como este são utilizados, conforme discutido no capítulo 2 desta pesquisa. É proposto neste trabalho uma nova base de dados de *emails* para que se obtenha a classificação entre *emails* legítimos e *spams*, de forma a auxiliar pesquisas mais atuais nesta área. Para a realização da base foram coletados 25745 *emails* entre o período de julho de 2014 e abril de 2017. Os *emails* foram capturados através do endereço <cemes@cemes.edu.br>.

O filtro antispam *SpamAssassin*, disponível em <http://spamassassin.apache.org/>, foi utilizado para a verificação de veracidade dos *emails*. O *SpamAssassin* é um filtro de *spam* de código aberto e amplamente utilizado. É composto de 711 testes que verificam a presença de palavras-chave contidas no corpo do *email*, testes no cabeçalho do *email* e também na *url* destes. Cada teste realizado pelo *SpamAssassin* é associado com uma pontuação. Caso contrário, se o teste realizado para verificação de spam seja satisfeito um valor numérico não nulo é atribuído a este teste. Caso contrário, se o teste realizado para verificação de *spam* seja satisfeito um valor numérico não nulo é atribuído a este teste.

A pontuação de todos os 711 testes é somada, e se para um determinado email a soma destes testes for maior ou igual que o limiar definido, por padrão o valor cinco, este email é definido como spam. Caso contrário, este e-mail é considerado como legítmo. É importante observar que estes valores não estão normalizados nesta etapa do processamento.

Dentre os testes realizados, nove destes, são associados com intervalos disjuntos de valores contínuos de saída de um classificador de texto (*naive bayes*). Os demais testes realizados pelo *SpamAssassin*, são para detectar características específicas do cabeçalho e do corpo do *email*, incluindo listas de bloqueio do DNS e bases de dados de filtros colaborativos. Valores padrões são fornecidos para as pontuações de cada teste e para o limiar de decisão. Estes valores podem ser alterados pelos usuários. Porém, por convenção os valores padrões foram utilizados neste trabalho, ou seja, cinco. O *SpamAssassin* utiliza um mecanismo SSl que automaticamente

retreina o classificador bayesiano utilizando emails não rotulados coletados durante a operação de filtragem. Em particular, o SpamAssassin utiliza emails em que a soma dos subconjuntos de testes são muito grandes (emails rotulados como *spam*) ou muito pequenas (emails rotulados como verídicos) do que os limiares definidos.

Logo, através da *API* disponível em <http://spamcheck.postmarkapp.com/> é possível carregar todos os *emails* coletados, e obter a pontuação dos 711 testes realizados pelo *SpamAssassin* para cada um dos *emails*. Após a obtenção dos resultados dos testes, este são organizados e é realizado a verificação quanto a autenticidade do *email*. A pontuação dos 711 testes realizados é somada vide figura disponível no Anexo A.

Conforme os testes realizados, dos 25745 *emails* coletados, foram classificados 7830 *emails* como *spams* e 17915 *emails* como legítimos.

Os dados obtidos foram normalizados na escala $[0,1]$ para considerar que as variáveis estejam nas mesmas proporções, isto facilita processo de classificação. A normalização unificada do i-ésimo elemento do k-ésimo vetor de dados é dada por:

$$x^r_{i(k)} = \frac{x_{i(k)} - \min\limits_{\forall j}(x_{i(j)})}{\max\limits_{\forall j}(x_{i(j)}) - \min\limits_{\forall j}(x_{i(j)})} \tag{4.1}$$

em que $j = 1, ..., k-1$; e $x^r_{i(k)} \in [0,1]$.

Contudo, a rotulação das classes foi obtida de forma manual, antes do processo de normalização, uma vez que, baseado no limiar padrão do SpamAssassin, se a soma dos testes realizados for maior ou igual a 5, a mensagem é considerada spam, do contrário um email legítmo.

4.2 Métodologia para classificação evolutiva

Os métodos utilizados neste trabalho são semi-supervisionados, ou seja, utilizam dados rotulados e não-rotulados para construir padrões em sistemas de classificação. Geralmente a aquisição de dados rotulados exigem conhecimento especialista para a classificação destes. A classificação manual pode não ser uma estratégia adequada quando considera-se grandes volumes de dados. Há situações em que as instancias são rotulados e aparentemente sugerem a utilização de métodos que utilizam aprendizagem supervisionada. No entanto, pode haver incertezas quanto ao processo de rotulação destas amostras. Neste casos os métodos que utilizam

aprendizagem semi-supervisionada utilizam apenas uma fração das instâncias que supostamente foram rotuladas corretamente.

Considerando um par de entrada e saída (x, y) respectivamente, onde $y = f(x)$. Procura-se uma aproximação para a função f que permita prever um valor para y dado um valor x. Em problemas de classificação, y é rótulo da classe, no caso deste trabalho utiliza-se duas classes $(C_!, C_2)$. Classificação de fluxo de dados envolve pares $(x, C)^{[h]}$ de dados sequenciados no tempo indexados por h. A não-estacionariedade existente nos dados requer a utilização de métodos classificadores evolutivos para a identificação das relações variantes no tempo em $f^{[h]}$.

Para as análises dos métodos classificadores FBeM e TEDA Class, considera-se uma matriz de confusão constituída de duas linhas e duas colunas. Esta matriz representa o número de verdadeiros-positivos (VP), falso-positivos (FP), verdadeiro-negativos (VN), e falso-negativos (FN), assim como no método ROC (Característica de Operador de Recepção). A previsão de acurácia dos métodos classificadores pode ser definida como:

$$Acuracia = \frac{VP + VN}{VP + FP + VN + FN} * 100\%, \tag{4.2}$$

Esta métrica de acurácia geralmente é empregada em bases de dados em que há um equilíbrio de classificação, caso da base proposta neste trabalho. O método ROC fornece um modo conveniente de se avaliar a qualidade dos classificadores evolutivos em ambientes que os dados são desconhecidos e não-estacionários, pois este método é insensível a mudanças nas distribuições das classes e as proporções de amostras por classes. O espaço ROC é definido através da taxa de VP e FP,

$$TaxaVP = \frac{VP}{VP + FN} \tag{4.3}$$

$$TaxaFP = \frac{FP}{FP + VN}, \tag{4.4}$$

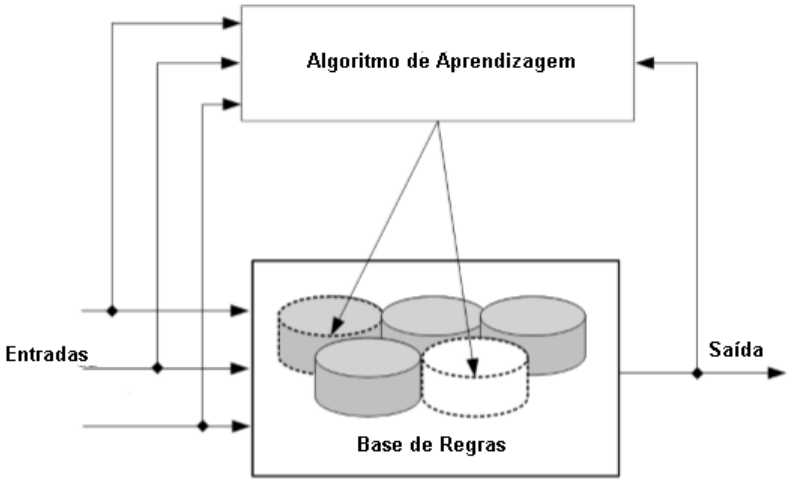
para cada classe, o método ROC aplica um valor limiar entre o intervalo $[0, 1]$ as saídas. Quanto mais uma curva for próximo ao vértice superior esquerdo, melhor é o resultado de classificação.

Os parâmetros inciais para os métodos classificadores foram definidos da seguinte forma: FBeM – $\rho = 0.06, h_r = 10000, \eta = 1$; TEDA Class – $m = 2$.

O método para seleção de características baseado em colônia de formigas artificiais (ACO), é utilizado para reduzir o número de características da base de dados. A base de dados é testada integralmente pelos métodos evolutivos, ou seja, com as 711 características originais como entrada dos modelos evolutivos. Entradas dos modelos evolutivos com 75%,50% e 25% do número total de características serão utilizadas também afim de se comparar a acurácia de classificação desses modelos. O número de regras criadas pelos modelos evolutivos, assim como o tempo de processamento gasto por estes para classificação destas amostras também são considerados nas análises.

A metodologia para classificação evolutiva da base de dados proposta é ilustrada no esquema apresentado na Figura 4.1.

Figura 4.1 – Esquema para classificação evolutiva



Fonte: Elaborado pelo autor

5 RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados os resultados das análises das simulações computacionais para classificação da base de dados de *emails*. São considerados os métodos evolutivos FBeM e TEDA Class.

5.1 Resultados das Simulações Computacionais

A base de dados obtidas neste trabalho é composta por 711 características diferentes obtidas conforme a metodologia descrita no capítulo anterior. A última caraterística desta base é dedicada aos rótulos da classes. A classe de rótulo 0 é classificada como *email* válido, e a classe de rótulo 1 é classificada como *spam*. Os rótulos são utilizados apenas para a verificação da acurácia de classificação dos modelos evolutivos, pois estes são semi-supervisionados. A base é constituída de 25745 amostras, sendo 7830 amostras classificadas como spams e 17915 amostras classificadas como *emails* legítimos.

No Apêndice A, é listado os 711 testes que são realizados em diferentes áreas do email, o Em alguns testes, a pontuação é atribuída à mensagem com valores negativos, indicando que tal característica pode sugerir que o a mensagem provém de uma fonte confiável. Porém, o fato de uma mensagem ter sido enviada de tal fonte, não garante que a mensagem não seja um spam. No teste 18, por exemplo, o valor -1 é atribuído à mensagem que não tenham sido enviadas de um servidor utilizando o protocolo SMTP e que pode indicar um servidor legítimo, contudo não garante que tal servidor não esteja sendo utilizado para a prática de spam. Contudo nos testes 297 a 303 o sistema verifica se no cabeçalho da mensagem possui uma chave codificada pelo emissor que deverá ser decodificada pelo receptor. O emissor, para atribuir tal chave no cabeçalho, gasta tempo e processamento para gerar um código que pode variar entre 20 e 25 bits, porém é facilmente decodificada pelo receptor. Uma vez que, para cada mensagem um bom tempo de processamento é gasto, para gerar a chave, dificilmente tal mensagem será oriunda de emails em massa. Em outros testes, o valor atribuído é próximo ao limiar cinco ou acima, indicando que a probabilidade da mensagem ser um spam é alta.

No teste 584, o sistema verifica se o nome do remetente contém nomes de drogas, caso o resultado seja positivo, a pontuação 4.29 é acrescida na mensagem. Também no teste 362, se o endereço do remetente da mensagem constar em uma lista negra, a pontuação atribuída à mesma será 100. Uma vez que o remete da mensagem está listado em uma lista de remetentes

de spam, a probabilidade de essa mensagem ser um spam obviamente é alta. A lista dos nomes dos testes, suas características e pontuação podem se verificadas no Apêndice A deste trabalho.

Simulações computacionais foram realizadas para avaliar a precisão dos métodos evolutivos FBeM e TEDA Class combinados com o método para ordernação e seleção de características baseado em colônia de formigas. A Tabela 5.1 sumariza os resultados obtidos pelos métodos evolutivos para a classificação da base de *emails* considerando diferentes números de características. O número de regras FBeM é delimitado pelo ρ, quanto menor o valor deste, mais grânulos são criados. Do mesmo modo o número de regras TEDA é limitado através do parâmetro m. Maior acurácia pode ser obtida conforme o número de grânulos e nuvens cresce, porém para preservar a característica de interpretabilidade os parâmetros iniciais foram ajustados para que a estrutura final dos modelos evolutivos não contivessem mais que 10 modelos locais (regras fuzzy ou nuvens).

A base de dados também foi testada em uma aplicação clássica que utiliza processamento offline a fim de comparar sua acurácia e tempo de processamento. A clusterização é um método de análise de dados utilizado para agrupar objetos em uma coleção com padrões que possuem algum tipo de similaridade. Trata-se de um processo realizado através da extração de dados com características comuns a um grupo. Visa-se identificar a qual grupo em uma série de grupos o elemento tenderá a pertencer. O método auxilia no processo de classificação dos dados, e é utilizado em aplicações para diversas áreas, como mineração de dados, processamento de imagens e tomadas de decisão, onde não existem informações exatas sobre os dados apresentados .

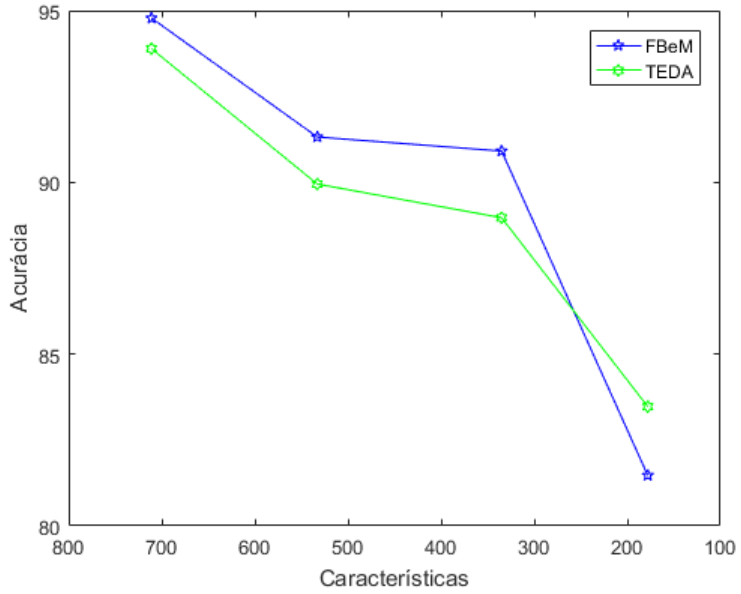
Tabela 5.1 – Classificação da base de dados de *email*

–	**FBeM**		
# características	Acurácia	Tempo(s)	# regras
711	**94,78**	137,23	10
533	91,32	100,12	7
355	90,91	89,69	7
178	81,47	**73,72**	7
–	**TEDA Class**		
# características	Acurácia	Tempo(s)	# nuvens
711	**93,89**	92,34	10
533	89,95	89,58	10
355	88,97	72,94	10
178	83,47	**57,33**	10
–	**Fuzzy C-Means**		
# características	Acurácia	Tempo(s)	# classes
711	**86,5**	294	2

À partir dos resultados obtidos na Tabela 5.1, nota-se que os modelos evolutivos são mais precisos quando todas as 711 características são consideradas como entrada destes. Isto ocorre pois algumas informações são perdidas quando algumas caracteristiscas não são utilizadas como entrada destes modelos. Os modelos baseados em regras fuzzy obtiveram uma maior acurácia do que os baseados em nuvens. No entanto, um grande número de características podem ser removidas sem uma perca significante acurácia conforme pode ser observado na Figura 5.1. O método para seleção de características utilizado neste trabalho desempenha um papel fundamental para a redução da complexidade computacional, como também para evitar "maldição da dimensionalidade"que é um problema crítico em um ambiente de fluxo de dados online.

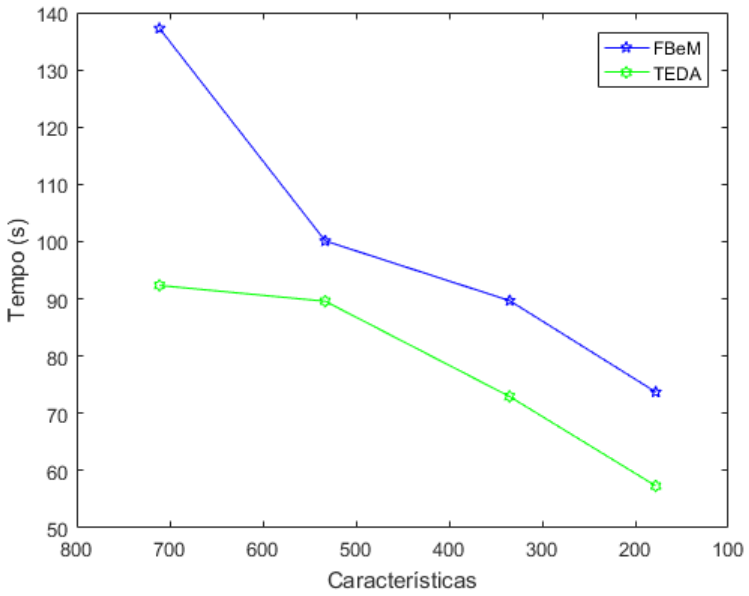
Conforme a Tabela 5.1 e ilustrado pela Figura 5.1, FBeM para classificação utilizando a totalidade da base de dados como entrada para a criação do modelo obteve acurácia = $94,78\%$. Utilizando o mesmo o número de entradas para a criação do modelo TEDA, o método baseado em tipicidade e excentricidade dos dados obteve acurácia = $93,89\%$. No entanto, nota-se um que o TEDA Class utiliza uma menor complexidade computacional para processar todas as amostras do fluxo de dados. Conforme ilustra a Figura 5.1, nota-se que apesar de um decréscimo linear na acurácia em ambos os métodos conforme as características são retiradas do espaço de entrada, esta perda de acurácia é compensada pelo decréscimo no tempo de processamento (Figura 5.2). O número de termos antecedentes das regras de ambos modelos, a compactação dos modelos resultantes, o tempo de processamento são aprimorados ao custo de

Figura 5.1 – Comparação de acurácia para diferentes números de características



Fonte: Elaborado pelo autor

Figura 5.2 – Comparação de $Tempo(s)$ para diferentes números de características



Fonte: Elaborado pelo autor

uma perca marginal de acurácia na classificação. Contudo, é importante lembrar que, em um ambiente real, um servidor MTA processa milhões de e-mails por segundo e, por isso, o tempo de processamento precisa ser considerado, uma vez que um segundo a mais ou a menos poderá fazer muita diferença. A capacidade de envio de cada servidor MTA é relativa ao tipo de serviço contratado, os servidores utilizados em planos mais simples tendem a ter limitações impostas

pelos provedores de serviço, pois em sua grande maioria são servidores compartilhados. Porém, se o servidor MTA for dedicado, ou seja, utilizado apenas para envio e recebimento de e-mails tende a processar um número maior de mensagens. Portanto, a longo prazo, o aumento ou diminuição do tempo de processamento pode impactar diretamente no desempenho bem como no valor gasto para manter um servidor MTA online.

No segundo experimento, com 75% do número total de características no espaço de entrada, o FBeM para classificação possuí uma acurácia de $91,32\%$, no entanto o tempo para processamento destas amostras decai de $137,23$ segundos para $100,12$ segundos. Ou seja, enquanto a acurácia tem uma perda de $3,65\%$, o tempo de processamento diminuiu $37,11$ segundos o mesmo que $27,45\%$. O TEDA Class com os mesmos 75% de características, apresentou uma diminuição de $4,19\%$ na acurácia, quanto ao tempo, houve um decréscimo de $2,76$ segundos correspondendo a $2,99\%$ do tempo total. Portanto, FBeM quando diminuído o número de características das amostras, apresentou menor perda de acurácia e, além disso, obteve melhoria considerável no tempo de processamento.

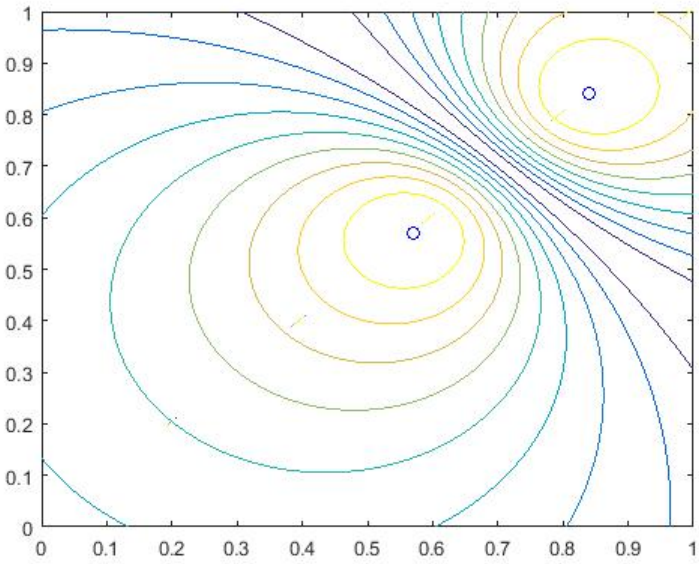
Com a redução de 50% do número de características, FBeM apresentou perda de $0,37\%$ na acurácia, quanto ao tempo de processamento, houve uma queda de $10,43$ segundos correspondendo a $10,4\%$ do tempo total. Neste caso, enquanto o tempo de processamento não teve o mesmo desempenho do experimento anterior, houve uma queda menor quanto à acurácia. No experimento com o TEDA Class, a acurácia diminuiu de $89,95\%$ pra $88,97\%$ ou seja, $1,09\%$, observou-se também, uma redução de $89,58$ segundos para $72,94$ segundos, no tempo de processamento, o que representa uma diminuição de $18,57\%$ do tempo total, comparado ao segundo experimento. Observou-se neste portnto, que FBeM tem uma menor perda de acurácia comparado ao TEDA Class, porém ambos os algoritmos obtiveram uma diminuição significativa no tempo de processamento.

No experimento seguinte, ao diminuir o número de características para 25% do total, FBeM apresentou uma diminuição de $10,38\%$ na acurácia e, quanto ao tempo de processamento, houve uma queda de $17,8\%$, comparados com os resultados do experimento anterior. O TEDA Class apresentou uma redução de $6,18\%$ na acurácia e $21,4\%$ no tempo de processamento. Percebeu se, neste caso, que FBeM, teve um desempenho inferior comparado ao TEDA Class. Quanto à acurácia, enquanto um teve uma redução de $90,91\%$ para $81,47\%$ o outro apresentou uma queda de $88,97\%$ para $83,47\%$. Neste caso, TEDA Class obteve, também, uma

diminuição maior do tempo total de processamento, ou seja, de $72,94$ segundos para $57,33$ segundos.

Para a realização do teste com o método Fuzzy C-Means, a base foi inicialmente normalizada e misturada, em seguida, utilizou se 70% dos dados para que FCM fosse treinado e definisse os centróides. Logo depois, foi utilizado os restante dos dados, ou seja, 30% para validação onde, o centro dos clusters dos dados de treinamento foram utilizados em comparação com os centros dos dados de validação. As classes obtidas na validaçao, foram comparadas às classes obtidas no treinamento.

Figura 5.3 – Cluster formados pelo método C-Means



Fonte: Elaborado pelo autor

Dos resultados obtidos, listados na Tabela 5.1, o tempo de processamento foi superior aos métodos online utilzados neste trabalho. FCM demorou 294 segundos para processar toda a base e apresentou acurácia de 86.5%. Além disso, em ambientes onde o fluxo de dados pode mudar a qualquer momento, faz se necessário, portanto, que parâmetros e estruturas possam ser ajustados a fim de garantir que a abordagem utilizada classifique corretamente os dados. Outro ponto importante é que, se o fluxo de dados é alterado, será necessario novos treinamentos e o conhecimento de um especialista para rotular novamente as classes obtidas. Este procedimento se torna inviável em uma aplicação real pois até que o algoritmo seja retreinado, o spam pode ter atingido grande número de usuários.

Tanto a acurácia quanto o tempo de processamento são variáveis importantes em um ambiente real, é preciso parcimônia para escolher a melhor relação entre um e outro. Além disso,

enquanto o acerto tem relação direta com a quantidade de spam que será entregue aos destinatários, o tempo que um servidor demora pra processar suas mensagens pode acarretar prejuízos, em empresas que necessitam de maior velocidade na entrega e recebimento das mensagens.

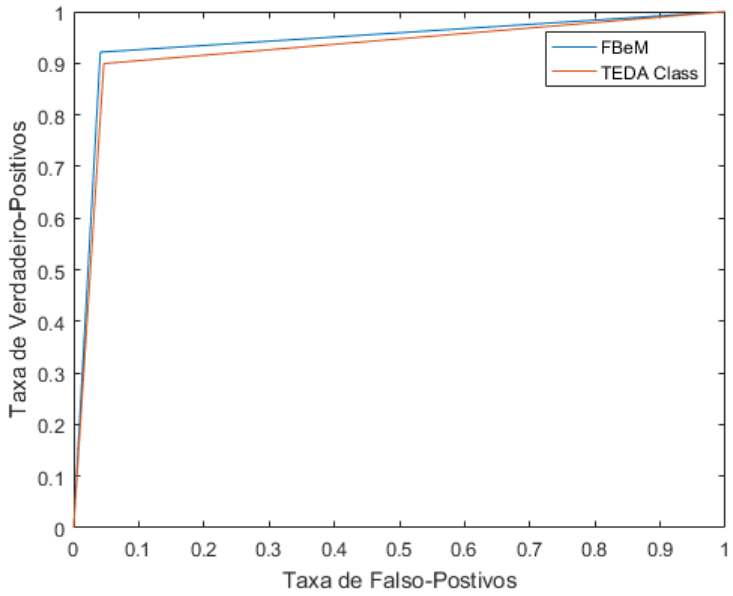
A próxima seção trata das análises das curvas ROC dos métodos classificadores evolutivos.

5.2 Análise de Curvas ROC

A análise da curva *Receiver operating characteristic* (ROC) fornece um método gráfico para avaliação, organização e seleção de sistemas de classificação. A curva ROC é particulamente útil para casos onde há um desequilíbrio entre as distribuições das classes.

As Figuras5.4 mostra a curva ROC para as classificações dadas pelos métodos FBeM e TEDA Class. Quanto mais a curva se aproxima do vértice esquerdo superior, melhor é a classificação das asmostras realizadas pelos métodos evolutivos. Note que a curva dada pelo método FBeM Class é mais próxima do vértice esquerdo superior do que a dada pelo método baseado em nuvens de dados.

Figura 5.4 – Curva ROC para os métodos FBeM e TEDA Class



Fonte: Elaborado pelo autor

Isto ocorre pois o método FBeM é capaz de produzir uma melhor classificação verdadeiros-positivos e menos falso-positivos que o método TEDA Class. Esta melhor capacidade de clas-

sificação é demonstrada na Tabela 5.1. Outra forma de análise dos resultados de classificação é obtida através da análise da matriz de confusão. A seção seguinte trata destas análises.

5.3 Matriz de Confusão

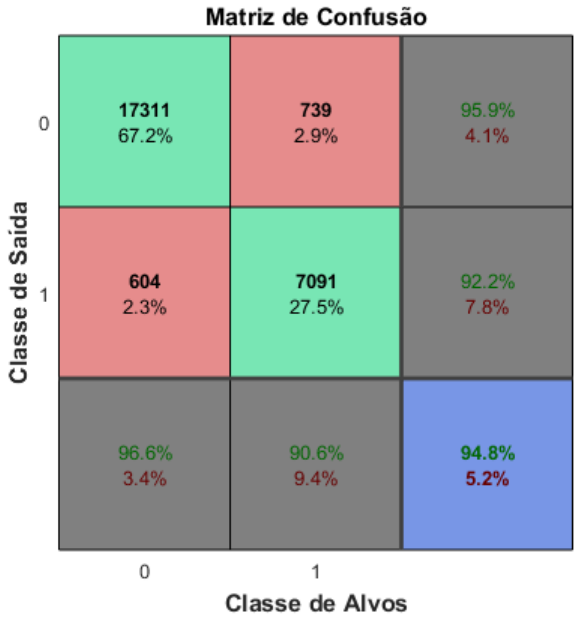
A matriz de confusão permite a visualização do desempenho de classificação dos métodos evolutivos utilizados. Cada coluna da matriz de confusão representa as instâncias previstas de uma determinada classe. Enquanto cada linha da matriz representa as instâncias na classe atual.

A Figura 5.5 ilustra a matriz de confusão dada pelo método FBeM em seu meu melhor caso de acurácia, ou seja, utilizando a totalidade da base de dados. A matriz de confusão indica um melhor desempenho de classificação do FBeM para a classe 0, portanto um melhor desempenho para classificar as amostras que são consideradas como *emails* válidos. A acurácia de classificação para esta classe foi de $96,6\%$. O desempenho de classificação do FBeM para instâncias da classe 1, ou seja, para a classe de *spams* foi de $90,6\%$. Isto ocorre pois há uma dificuldade maior em determinar as as características que compõem um *spam*.

Assim como o método FBeM, a matriz de confusão para o método TEDA Class (Figura 5.6) revelou um melhor desempenho de classificação para as amostras consideradas da classe 0, ou seja, amostras de *emails* válidos. Classificando $95,7\%$ destas corretamente. A acurácia para as instâncias da classe 1 foi de $89,7\%$. Ambos métodos mostraram maior dificuldade em determinar os padrões que compoõem a classe *spam*. Isto ocorre por conta das diversas mudanças que pode haver na estrutura de um *spam*.

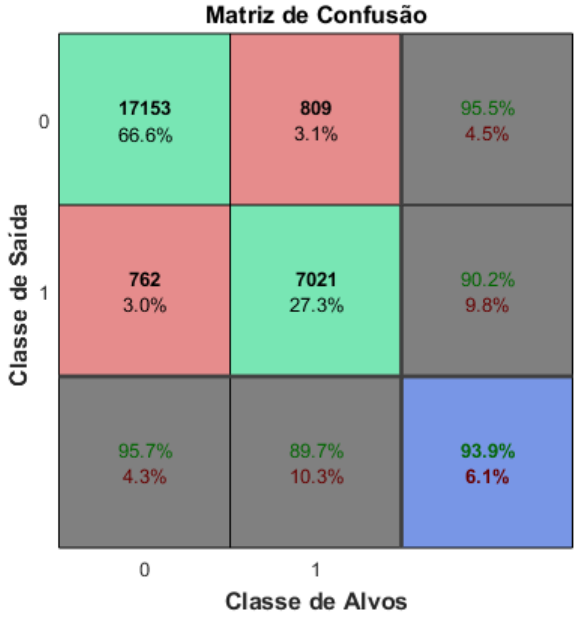
Os resultados obtidos mostram que os métodos evolutivos obtiveram uma boa acurácia nos testes realizados. Em ambientes em que a fonte geradora de dados mudam de forma abrupta, como o é o caso de *spams*, métodos convencionais podem não ser indicados. Além do mais por se tratar de fluxo de dados online. FBeM obteve um melhor resultado de classificação nos três primeiros testes, ou seja, com 100%, 75% e 50% das características. Além disso, verificou se que, ao diminuir a dimensão da base de e-mails o tempo de processamento diminuiu consideravelmente. Esta característica pode ser explorada em servidores MTAs onde o processamento e a memória são compartilhados com outros servidores, também onde a velocidade de envio e recebimento de e-mails é desejada. TEDA Class apresentou maior perda de acurácia, quanto diminuído o número de características, contudo obteve melhor índice de acerto, no quarto experimento, onde foi subtraído 75% das variáveis. Neste caso, Teda apresentou uma redução de

Figura 5.5 – Matriz de confusão da classificação dada pelo método FBeM



Fonte: Elaborado pelo autor

Figura 5.6 – Matriz de confusão da classificação dada pelo método TEDA Class



Fonte: Elaborado pelo autor

6,18% enquanto FBeM apresentou uma queda de 10,38% na acurácia. Além disso, tanto FBeM quanto TEDA Class tiveram redução semelhantes do tempo de processamento, 15,97 segundos e 15,61, concomitantemente.

Apesar disso, é importante refletir sobre o que é mais importante para um sistema classificador de emails, se quanto à acurácia ou tempo de processamento. Se por um lado a acurácia para spams foi de 90,6%, é importante salientar que, em uma aplicação real, é um numero bastante razoável. Reduzir em 10% o volume de mensagens de spam que trafegam na rede representa ganhos de armazenamento, produtividade, menor risco de segurança, menor utilização de tráfego de dados, etc. Por outro lado, o modelo conseguiu acertar 96,6% da classe de e-mails legítimos, isto representa um ótimo resultado do ponto de vista de uma aplicação real, uma vez que somente cerca de 4% das mensagens seriam classificadas incorretamente. Muitas mensagens de e-mails legítimas possuem características que se confundem com mensagens de spam. Por exemplo, se o usuário assinou uma lista de emails de um site específico, este por sua vez enviará mensagens contendo promoções, imagens, HTML, o que pode confundir os mecanismos de classificação, pois tais características também estão presentes em mensagens de spam. Contudo, é mais vantajoso que o sistema consiga classificar corretamente um maior numero de mensagens de emails legítimos do que o contrário, embora o usuário possa ser exposto a um número maior de spams, tais mensagens podem ser classificadas pelo próprio utilizador do sistema. Se a mensagem legítima for classificada como spam, o usuário pode não ser capaz de identificar. Isto por que a grande maioria dos sistemas de e-mails mantém uma pasta específica pra mensagens de spams, movendo tais mensagens automaticamente e inclusive excluindo-as depois de um certo tempo. Assim como o método FBeM, a matriz de confusão para o método TEDA Class (Figura 5.6) revelou um melhor desempenho de classificação para as amostras consideradas da classe 0, ou seja, amostras de emails válidos. Classificando 95,7% destas corretamente. A acurácia para as instâncias da classe 1 foi de 89,7%.

Ambos os métodos mostraram maior dificuldade em determinar os padrões que compõem a classe spam. Isto ocorre por conta das diversas mudanças que incidem durante o fluxo de dados, pois cada mensagem pode vim de um emissor diferente e utilizando técnicas diferentes.

A escolha do método ideal então depende da aplicação em que este está inserido. Em ambientes em que o tempo não é uma restrição, pode se optar por uma maior precisão de acurácia. No entanto, o tempo de processamento pode ser uma restrição em ambientes em há um fluxo de dados de alta frequência, com alta dimensionalidade e também em ambientes onde a informação necessita de maior velocidade entre o emissor e o receptor.

6 CONCLUSÃO

O envio de *spams* através de *emails* tem causado vários problemas á usuários. Problemas que vão desde o aborrecimento de usuários, que encontram dificuldade devido ao acúmulo de mensagens irrelevantes em suas caixas de entrada, até ao ponto de sobrecarregar um servidor SMTP, devido ao grande volume de mensagens. Em muitos casos, o dano excede gravemente os motivos citados acima chegando ao ponto de os dados dos usuários ou empresas serem comprometidos e gerar prejuízos morais e financeiros.

Este trabalho propôs a criação de uma nova base de dados para a detecção de *spams*. Haja visto que a base amplamente utilizada pela literatura não corresponde mais as atuais técnicas de envio de *spams* utilizada pelos *spammers*. A base proposta possuí 712 características, sendo à última dedicada aos rótulos de classificação à serem utilizados pelos classificadores. Das 25745 amostras que compõem a base de dados, 7830 amostras são *spams* e 17915 amostras são *emails* legítimos. Os dados foram coletados entre o período de julho de 2014 e abril de 2017.

Para classificar os dados, os métodos inteligentes evolutivos FBeM e uma variação do método evolutivo baseado em tipicidade e excentricidade dos dados, TEDA, foram aplicados. Um método de classificação e seleção de características baseados em colônia de formigas artificiais foi utilizado para identificar o melhor conjunto de características a serem usados pelos modelos classificadores evolutivos. O método reduziu a complexidade computacional global e acelerou as etapas de processamento.

De modo geral, os experimentos demonstraram que os métodos evolutivos são eficientes para classificar as amostras. Os resultados revelam que ambos métodos evolutivos mostraram uma dificuldade maior para classificar as instâncias classificadas como *spams*. Isto ocorre devido à mudanças abruptas que ocorrem na fonte geradora destes dados, ou seja, as diversas técnicas utilizadas pelos *spammers* para envio destes, que estão em constante mudança. O modelo baseado em conjuntos fuzzy, FBeM, mostrou melhores índices de acurácia nos testes realizados, como revelam as análises das curvas *ROC* e matriz de confusão.

O modelo baseado na excentricidade dos dados, TEDA Class, mostrou uma menor complexidade computacional para classificação das instâncias. O tempo de processamento pode ser um problema crítico em aplicações com fluxo de dados de alta frequência e alta dimensão. Portanto, para cada situação uma solução diferente é recomendada. Neste caso isto irá depender da frequência com que os *emails* chegam.

Trabalhos futuros irão discutir novas características que podem ser acrescentadas à bases de dados, além de diferentes valores de limiares para definirem a classificação das amostras quanto serem *spam* ou não. Trabalhos futuros devem considerar também a seleção incremental de características e diferentes variáveis estatísticas associadas as regras fuzzy e nuvens. Como por exemplo, informações sobre especificidade, entropia, correntropia e cardinalidade associadas à estas. Estas variáveis podem ser úteis na otimização estrutural dos modelos durante o aprendizado *online*. Processamento de dados heterogêneos, ou seja, misturas de dados numéricos, intervalares, fuzzy e intervalos fuzzy, é também um tópico possível de pesquisa. A análise de spam em imagens anexas é outro tópico de investigação possível.

REFERÊNCIAS

ABRAHAM, A.; GROSAN, C.; PEDRYCZ, W. **Engineering evolutionary intelligent systems**. [S.l.]: Springer, 2008. v. 82.

AGHDAM, M. H.; GHASEM-AGHAEE, N.; BASIRI, M. E. Text feature selection using ant colony optimization. **Expert systems with applications**, Elsevier, v. 36, n. 3, p. 6843–6853, 2009.

ALLEN, J. D. et al. **The Unicode Standard**. [S.l.]: Citeseer, 2007.

ANGELOV, P. Evolving fuzzy systems. **Encyclopedia of Complexity and Systems Science**, Springer, p. 3242–3255, 2009.

ANGELOV, P. Anomaly detection based on eccentricity analysis. In: IEEE. **Evolving and Autonomous Learning Systems (EALS), 2014 IEEE Symposium on**. [S.l.], 2014. p. 1–8.

ANGELOV, P. et al. **Evolving fuzzy systems**. [S.l.]: IEEE Press, 2006.

ANGELOV, P. P. **Evolving rule-based models: a tool for design of flexible adaptive systems**. [S.l.]: Physica, 2013. v. 92.

ANGELOV, P. P.; ZHOU, X. Evolving fuzzy-rule-based classifiers from data streams. **IEEE Transactions on Fuzzy Systems**, IEEE, v. 16, n. 6, p. 1462–1475, 2008.

ÅSTRÖM, K. J.; WITTENMARK, B. **Adaptive control**. [S.l.]: Courier Corporation, 2013.

BIANCHI, N. **The Return of the Open Relays**. 2015.

BIGGIO, B. et al. Image spam filtering by content obscuring detection. In: **CEAS**. [S.l.: s.n.], 2007.

BISHOP, C. M. **Pattern recognition and machine learning**. [S.l.]: springer, 2006.

BRAGA, Í. A.; LADEIRA, M. Um modelo adaptativo para a filtragem de spam. In: **VI Encontro Nacional de Inteligência Artificial, Rio de Janeiro–RJ, Anais do XXVII Congresso da Sociedade Brasileira de Computação**. [S.l.: s.n.], 2007. p. 1381–1390.

CALLAS, J. et al. **OpenPGP message format**. [S.l.], 2007.

CHEN, C.; TIAN, Y.; ZHANG, C. Spam filtering with several novel bayesian classifiers. In: IEEE. **Pattern Recognition, 2008. ICPR 2008. 19th International Conference on**. [S.l.], 2008. p. 1–4.

COCKERHAM, R. There are 600,426,974,379,824,381,952 ways to spell viagra. **Recuperado el**, v. 27, 2004.

COSTA, B. S. J.; ANGELOV, P. P.; GUEDES, L. A. Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier. **Neurocomputing**, Elsevier, v. 150, p. 289–303, 2015.

CROCKER, D. Rfc 822: standard for the format of arpa internet text messages; august 13, 1982. **See also STD0011. Obsoletes RFC0733. Updated by RFC1123, RFC1138, RFC1148, RFC1327, RFC2156. Status: STANDARD**, 1982.

DAMIANI, E. et al. P2p-based collaborative spam detection and filtering. In: IEEE. **Peer-to-Peer Computing, 2004. Proceedings. Proceedings. Fourth International Conference on**. [S.l.], 2004. p. 176–183.

DIMMOCK, N.; MADDISON, I. Peer-to-peer collaborative spam detection. **Crossroads**, ACM, v. 11, n. 2, p. 4–4, 2004.

DORIGO, M.; BIRATTARI, M.; STUTZLE, T. Ant colony optimization. **IEEE computational intelligence magazine**, IEEE, v. 1, n. 4, p. 28–39, 2006.

DRUCKER, H.; WU, D.; VAPNIK, V. N. Support vector machines for spam categorization. **IEEE Transactions on Neural networks**, IEEE, v. 10, n. 5, p. 1048–1054, 1999.

DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern classification**. [S.l.]: John Wiley & Sons, 2012.

FOSSI, M. et al. Symantec internet security threat report trends for 2010. **Volume XVI**, 2011.

FRANK, E.; HALL, M.; PFAHRINGER, B. Locally weighted naive bayes. In: MORGAN KAUFMANN PUBLISHERS INC. **Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence**. [S.l.], 2002. p. 249–256.

FREED, N.; BORENSTEIN, N. **Multipurpose internet mail extensions (MIME) part one: Format of internet message bodies**. [S.l.], 1996.

FRITZKE, B. Supervised learning with growing cell structures. In: **Advances in neural information processing systems**. [S.l.: s.n.], 1994. p. 255–262.

GAO, Y. et al. Image spam hunter. In: IEEE. **Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on**. [S.l.], 2008. p. 1765–1768.

GOLDBERG, D.; DEB, K.; KORB, B. Messy genetic algorithms: Motivation, analysis, and first results. **Complex systems**, n. 3, p. 493–530, 1989.

GRANT, I. H. Recursive least squares. **Teaching Statistics**, Wiley Online Library, v. 9, n. 1, p. 15–18, 1987.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. **Journal of machine learning research**, v. 3, n. Mar, p. 1157–1182, 2003.

GUZELLA, T. S.; CAMINHAS, W. M. A review of machine learning approaches to spam filtering. **Expert Systems with Applications**, Elsevier, v. 36, n. 7, p. 10206–10222, 2009.

HOANCA, B. How good are our weapons in the spam wars? **IEEE Technology and Society magazine**, IEEE, v. 25, n. 1, p. 22–30, 2006.

JAKOBSSON, M. Modeling and preventing phishing attacks. In: **Financial Cryptography**. [S.l.: s.n.], 2005. v. 5.

KABIR, M.; SHAHJAHAN, M.; MURASE, K. Ant colony optimization toward feature selection. In: BARBOSA, H. J. (Ed.). **Ant Colony Optimization - Techniques and Applications**. Rijeka: InTech, 2013. cap. 01. Disponível em: <http://dx.doi.org/10.5772/51707>.

KANAN, H.; FAEZ, K.; TAHERI, S. Feature selection using ant colony optimization (aco): a new method and comparative study in the application of face recognition system. **Advances in Data Mining. Theoretical Aspects and Applications**, Springer, p. 63–76, 2007.

KANGIN, D.; ANGELOV, P. Evolving clustering, classification and regression with teda. In: IEEE. **Neural Networks (IJCNN), 2015 International Joint Conference on**. [S.l.], 2015. p. 1–8.

KANGIN, D.; ANGELOV, P. New autonomously evolving classifier teda class. In: **IEEE International Joint Conference on Neural Networks, IJCNN-2015**. [S.l.: s.n.], 2015.

KASABOV, N. Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, IEEE, v. 31, n. 6, p. 902–918, 2001.

KASABOV, N. **Evolving connectionist systems: the knowledge engineering approach**. [S.l.]: Springer Science & Business Media, 2007.

KASABOV, N.; FILEV, D. Evolving intelligent systems: methods, learning, & applications. In: IEEE. **Evolving Fuzzy Systems, 2006 International Symposium on**. [S.l.], 2006. p. 8–18.

KASABOV, N. K. **Foundations of neural networks, fuzzy systems, and knowledge engineering**. [S.l.]: Marcel Alencar, 1996.

KASABOV, N. K.; SONG, Q. Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. **IEEE transactions on Fuzzy Systems**, IEEE, v. 10, n. 2, p. 144–154, 2002.

KIRAN, S. R. et al. Spam or not spam–that is the question. Citeseer, 2009.

KIRITCHENKO, S.; MATWIN, S. Email classification with co-training. In: IBM CORP. **Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research**. [S.l.], 2011. p. 301–312.

KLENSIN, J. C. Simple mail transfer protocol. 2008.

KOLEV, D. et al. Arfa: automated real-time flight data analysis using evolving clustering, classifiers and recursive density estimation. In: IEEE. **Evolving and Adaptive Intelligent Systems (EAIS), 2013 IEEE Conference on**. [S.l.], 2013. p. 91–97.

KOZA, J. R. **Genetic programming: on the programming of computers by means of natural selection**. [S.l.]: MIT press, 1992. v. 1.

LEITE, D. et al. Evolving fuzzy granular modeling from nonstationary fuzzy data streams. **Evolving Systems**, Springer, v. 3, n. 2, p. 65–79, 2012.

LEITE, D. et al. Evolving granular fuzzy model-based control of nonlinear dynamic systems. **IEEE Transactions on Fuzzy Systems**, IEEE, v. 23, n. 4, p. 923–938, 2015.

LEITE, D. F.; COSTA, P.; GOMIDE, F. Evolving granular classification neural networks. In: IEEE. **Neural Networks, 2009. IJCNN 2009. International Joint Conference on**. [S.l.], 2009. p. 1736–1743.

LEVINE, J. R. Experiences with greylisting. In: **CEAS**. [S.l.: s.n.], 2005.

LI, P. et al. Integration of local and global features for image spam filtering. **Journal of Computational Information Systems**, v. 8, n. 2, p. 779–789, 2012.

LINN, J. Privacy enhancement for internet electronic mail: Part i: message encryption and authentication procedures. 1993.

LIU, C.; STAMM, S. Fighting unicode-obfuscated spam. In: ACM. **Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit**. [S.l.], 2007. p. 45–59.

LUGHOFER, E. **Evolving fuzzy systems-methodologies, advanced concepts and applications**. [S.l.]: Springer, 2011. v. 53.

LUGHOFER, E.; ANGELOV, P. Handling drifts and shifts in on-line data streams with evolving fuzzy systems. **Applied Soft Computing**, Elsevier, v. 11, n. 2, p. 2057–2068, 2011.

MACIEL, L. et al. Evolving fuzzy systems for pricing fixed income options. **Evolving Systems**, Springer, v. 3, n. 1, p. 5–18, 2012.

NEMATI, S. et al. A novel aco–ga hybrid algorithm for feature selection in protein function prediction. **Expert systems with applications**, Elsevier, v. 36, n. 10, p. 12086–12094, 2009.

OLIVEIRA, J. V. D.; PEDRYCZ, W. **Advances in fuzzy clustering and its applications**. [S.l.]: John Wiley & Sons, 2007.

OLIVO, C. K.; SANTIN, A. O.; OLIVEIRA, L. E. S. Abordagens para detecção de spam de e-mail. 2010.

PEDRYCZ, W.; SKOWRON, A.; KREINOVICH, V. **Handbook of granular computing**. [S.l.]: John Wiley & Sons, 2008.

RAMSDELL, B. Secure/multipurpose internet mail extensions (s/mime) version 3.1 message specification. 2004.

SCHNEIDER, K.-M. A comparison of event models for naive bayes anti-spam e-mail filtering. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1**. [S.l.], 2003. p. 307–314.

SEIGNEUR, J.-M. et al. Combating spam with tea (trustworthy email addresses). In: **Proceedings of the Second Annual Conference on Privacy, Security and Trust (PST'04)**. [S.l.: s.n.], 2004. p. 47–58.

ŠKRJANC, I. Evolving fuzzy-model-based design of experiments with supervised hierarchical clustering. **IEEE Transactions on Fuzzy Systems**, IEEE, v. 23, n. 4, p. 861–871, 2015.

SOARES, E. et al. Cloud-based evolving intelligent method for weather time series prediction. In: IEEE. **Fuzzy Systems (FUZZ), 2017 IEEE International Conference on**. [S.l.], 2017.

SORANAMAGESWARI, M.; MEENA, C. A novel approach towards image spam classification. **International Journal of Computer Theory and Engineering**, IACSIT Press, v. 3, n. 1, p. 84, 2011.

WHITWORTH, B.; WHITWORTH, E. Spam and the social-technical gap. **Computer**, IEEE, v. 37, n. 10, p. 38–45, 2004.

XU, C. et al. Fusion of text and image features: A new approach to image spam filtering. In: **Practical Applications of Intelligent Systems**. [S.l.]: Springer, 2011. p. 129–140.

YAO, X.; LIU, Y. Fast evolutionary programming. **Evolutionary Programming**, v. 3, p. 451–460, 1996.

APÊNDICE A – Testes realizados pelos *SpamAssassin*

As Figuras 1 e 2 ilustram o teste realizado pelo *SpamAssassin* em um corpo de *email* e um teste utilizando a *API* descrita na seção de metodologia respectivamente.A Figura 3 ilustra a descrição de alguns testes realizados pelo *SpamAssassin*.

Figura 1 – Teste de *email* utilizando o *SpamAssassin*

```
Return-path: <super@emailvpis.com.br>
Envelope-to: alexandre@cemes.edu.br
Delivery-date: Tue, 28 Jul 2015 22:48:17 -0300
Received: from whost-01.parc01.emailvpis.com.br
([187.103.108.194]:58241)
        by eurol.euroti.com.br with esmtp (Exim 4.85)
        (envelope-from <super@emailvpis.com.br>)
        id 1ZKGTZ-0006w6-9D
        for alexandre@cemes.edu.br; Tue, 28 Jul 2015 22:48:17 -0300
Received: from www.emailvpis.com.br (unknown [127.0.0.1])
        by whost-01.parc01.emailvpis.com.br (Postfix) with ESMTPA id
714151A676642
        for <alexandre@cemes.edu.br>; Tue, 28 Jul 2015 22:49:15 -0300
(BRT)
To: alexandre@cemes.edu.br
Message-ID: <6adffe98d5437171083613f6095d22b2@www.emailvpis.com.br>
Date: Tue, 28 Jul 2015 11:14:57 -0300
From: "Elizangela" <super@emailvpis.com.br>
Reply-To: super@emailvpis.com.br
MIME-Version: 1.0
X-Mailer-LID: 5,7
List-Unsubscribe:
<http://www.emailvpis.com.br/envio/unsubscribe.php?M=2275442&C=d592652
d212f85falce5c2c8393e56cf&L=7&N=29>
X-Mailer-RecptId: 2275442
X-Mailer-SID: 29
X-Mailer-Sent-By: 1
Content-Type: multipart/alternative; charset="UTF-8";
boundary="b1_e8012a758aacc5107220d2f0a9626a33"
Content-Transfer-Encoding: 8bit
X-Spam-Status: Yes, score=7.1 ( Aqui o spamassassin gravou o score
final da mensagem ou seja 7.1)
X-Spam-Score: 71
X-Spam-Bar: +++++++
X-Spam-Report: Spam detection software, running on the system
"eurol.euroti.com.br",
 has identified this incoming email as possible spam.  The original
 message has been attached to this so you can view it or label
 similar future email.  If you have any questions, see
 root\@localhost for details.
```

Fonte: Elaborado pelo autor

Tabela 1 – Testes realizados

Lista	Pontuação	Nome do teste	Área testada
1	2.026	TRACKER_ID	body
2	0.001	WEIRD_QUOTING	body
3	1	EMAIL_ROT13	body
4	2.246	MPART_ALT_DIFF	body
5	2.799	MPART_ALT_DIFF_COUNT	body

6	1	BLANK_LINES_80_90	body
7	1	MULTIPART_ALT_NON_TEXT	body
8	3.200	CHARSET_FARAWAY	body
9	0.001	MIME_BASE64_BLANKS	rawbody
10	0.001	MIME_BASE64_TEXT	rawbody
11	0.001	MISSING_MIME_HB_SEP	body
12	0.354	MIME_HTML_MOSTLY	body
13	2.199	MIME_HTML_ONLY	body
14	0.001	MIME_QP_LONG_LINE	rawbody
15	1	MIME_BAD_ISO_CHARSET	body
16	1	HTTPS_IP_MISMATCH	body
17	0.001	URI_TRUNCATED	body
18	-1.000	ALL_TRUSTED	header
19	-0.001	NO_RELAYS	header
20	0	RCVD_IN_NJABL_RELAY	header
21	0	RCVD_IN_NJABL_SPAM	header
22	1	RCVD_IN_NJABL_MULTI	header
23	1	RCVD_IN_NJABL_CGI	header
24	0	RCVD_IN_NJABL_PROXY	header
25	0	RCVD_IN_SORBS_HTTP	header
26	0	RCVD_IN_SORBS_SOCKS	header
27	1	RCVD_IN_SORBS_MISC	header
28	1	RCVD_IN_SORBS_SMTP	header
29	0	RCVD_IN_SORBS_WEB	header
30	1	RCVD_IN_SORBS_BLOCK	header
31	1	RCVD_IN_SORBS_ZOMBIE	header
32	0	RCVD_IN_SORBS_DUL	header
33	0	RCVD_IN_SBL	header
34	0	RCVD_IN_XBL	header
35	0	RCVD_IN_PBL	header
36	0	DNS_FROM_RFC_DSN	header
37	0	DNS_FROM_RFC_BOGUSMX	header

38	0	DNS_FROM_AHBL_RHSBL	header
39	0	RCVD_IN_BL_SPAMCOP_NET	header
40	1	RCVD_IN_MAPS_RBL	header
41	1	RCVD_IN_MAPS_DUL	header
42	1	RCVD_IN_MAPS_RSS	header
43	1	RCVD_IN_MAPS_OPS	header
44	1	RCVD_IN_MAPS_NML	header
45	0	RCVD_IN_IADB_VOUCHED	header
46	2.108	SUBJECT_DRUG_GAP_C	header
47	2.799	SUBJECT_DRUG_GAP_L	header
48	1	SUBJECT_DRUG_GAP_S	header
49	1	SUBJECT_DRUG_GAP_VA	header
50	1	SUBJECT_DRUG_GAP_X	header
51	1	DRUG_DOSAGE	body
52	2.799	DRUG_ED_CAPS	body
53	0.001	DRUG_ED_SILD	body
54	1	DRUG_ED_GENERIC	body
55	0.696	DRUG_ED_ONLINE	body
56	0.843	ONLINE_PHARMACY	body
57	1.915	NO_PRESCRIPTION	body
58	1	VIA_GAP_GRA	body
59	3.300	DRUGS_SMEAR1	body
60	1.887	FAKE_HELO_MAIL_COM_DOM	header
61	1	HELO_DYNAMIC_ROGERS	header
62	2.629	HELO_DYNAMIC_DIALIN	header
63	2.321	HELO_DYNAMIC_HEXIP	header
64	3.031	HELO_DYNAMIC_SPLIT_IP	header
65	2.815	HELO_DYNAMIC_IPADDR2	header
66	2.412	HELO_DYNAMIC_CHELLO_NL	header
67	2.385	HELO_DYNAMIC_HOME_NL	header
68	0.001	FREEMAIL_FROM	header
69	2.602	FREEMAIL_ENVFROM_END_DIGIT	header

70	1.221	FREEMAIL_REPLYTO_END_DIGIT	header
71	1	FRAGMENTED_MESSAGE	header
72	2.099	FROM_BLANK_NAME	header
73	2.801	FROM_STARTS_WITH_NUMS	header
74	2.699	FROM_OFFERS	header
75	0.001	FROM_NO_USER	header
76	2.366	MSGID_SPAM_CAPS	header
77	1	MSGID_SPAM_LETTERS	header
78	0.797	MSGID_YAHOO_CAPS	header
79	0.001	MSGID_SHORT	header
80	0.001	MSGID_MULTIPLE_AT	header
81	1	DATE_SPAMWARE_Y2K	header
82	1.701	INVALID_DATE	header
83	0.262	INVALID_DATE_TZ_ABSURD	header
84	1	INVALID_TZ_CST	header
85	1	INVALID_TZ_EST	header
86	0.953	ENGLISH_UCE_SUBJECT	header
87	1	JAPANESE_UCE_SUBJECT	header
88	1	KOREAN_UCE_SUBJECT	header
89	2.499	FORGED_TELESP_RCVD	header
90	1	NONEXISTENT_CHARSET	header
91	1	PREVENT_NONDELIVERY	header
92	0.001	X_IP	header
93	2.711	SUBJ_AS_SEEN	header
94	0.600	SUBJ_DOLLARS	header
95	3.299	SUBJ_YOUR_DEBT	header
96	2.910	SUBJ_YOUR_FAMILY	header
97	2.799	RCVD_FAKE_HELO_DOTCOM	header
98	1.927	SUBJECT_DIET	header
99	1.0	EXTRA_MPART_TYPE	header
100	3.016	MIME_BOUND_DD_DIGITS	header
101	0.432	MIME_BOUND_DIGITS_15	header

102	1	MIME_BOUND_MANY_HEX	header
103	0.892	TO_MALFORMED	header
104	1	WITH_LC_SMTP	header
105	0.594	SUBJ_BUY	header
106	1	RCVD_AM_PM	header
107	1	FAKE_OUTBLAZE_RCVD	header
108	2.699	UNCLOSED_BRACKET	header
109	0.500	FROM_DOMAIN_NOVOWEL	header
110	0.500	FROM_LOCAL_NOVOWEL	header
111	0.000	FROM_LOCAL_HEX	header
112	0.001	FROM_LOCAL_DIGITS	header
113	1	X_PRIORITY_CC	header
114	3.099	BAD_ENC_HEADER	header
115	3.399	RCVD_ILLEGAL_IP	header
116	3.200	CHARSET_FARAWAY_HEADER	header
117	2.192	FROM_ILLEGAL_CHARS	header
118	1	HEAD_ILLEGAL_CHARS	header
119	0.001	FORGED_HOTMAIL_RCVD2	header
120	2.397	FORGED_YAHOO_RCVD	header
121	1.801	SORTED_RECIPS	header
122	2.499	SUSPICIOUS_RECIPS	header
123	0.915	MISSING_HEADERS	header
124	2.399	DATE_IN_PAST_03_06	header
125	1.699	DATE_IN_PAST_06_12	header
126	0.001	DATE_IN_PAST_12_24	header
127	1.109	DATE_IN_PAST_24_48	header
128	2.600	DATE_IN_PAST_96_XX	header
129	3.399	DATE_IN_FUTURE_03_06	header
130	2.899	DATE_IN_FUTURE_06_12	header
131	2.603	DATE_IN_FUTURE_12_24	header
132	2.598	DATE_IN_FUTURE_24_48	header
133	2.384	DATE_IN_FUTURE_48_96	header

134	2.614	DATE_IN_FUTURE_96_XX	header
135	3.035	UNRESOLVED_TEMPLATE	header
136	0.518	SUBJ_ALL_CAPS	header
137	0.001	LOCALPART_IN_SUBJECT	header
138	3.899	MSGID_OUTLOOK_INVALID	header
139	1	HEADER_COUNT_CTYPE	header
140	1	HEAD_LONG	header
141	1	MISSING_HB_SEP	header
142	0.001	UNPARSEABLE_RELAY	header
143	1.680	RCVD_HELO_IP_MISMATCH	header
144	0.001	RCVD_NUMERIC_HELO	header
145	3.100	NO_RDNS_DOTCOM_HELO	header
146	0.001	HIDE_WIN_STATUS	rawbody
147	0.001	HTML_MESSAGE	body
148	1	HTML_COMMENT_SHORT	body
149	0.198	HTML_COMMENT_SAVED_URL	body
150	0.001	HTML_EMBEDS	body
151	0.001	HTML_EXTRA_CLOSE	body
152	0.001	HTML_FONT_SIZE_LARGE	body
153	0.001	HTML_FONT_SIZE_HUGE	body
154	0.713	HTML_FONT_LOW_CONTRAST	body
155	0.001	HTML_FONT_FACE_BAD	body
156	1	HTML_FORMACTION_MAILTO	body
157	1.680	HTML_IMAGE_ONLY_04	body
158	0.585	HTML_IMAGE_ONLY_08	body
159	1.381	HTML_IMAGE_ONLY_12	body
160	1.969	HTML_IMAGE_ONLY_16	body
161	2.109	HTML_IMAGE_ONLY_20	body
162	2.799	HTML_IMAGE_ONLY_24	body
163	2.799	HTML_IMAGE_ONLY_28	body
164	2.196	HTML_IMAGE_ONLY_32	body
165	2.199	HTML_IMAGE_RATIO_02	body

166	2.089	HTML_IMAGE_RATIO_04	body
167	0.001	HTML_IMAGE_RATIO_06	body
168	0.001	HTML_IMAGE_RATIO_08	body
169	0.601	HTML_OBFUSCATE_05_10	body
170	0.174	HTML_OBFUSCATE_10_20	body
171	2.499	HTML_OBFUSCATE_20_30	body
172	1	HTML_OBFUSCATE_30_40	body
173	1	HTML_OBFUSCATE_50_60	body
174	1	HTML_OBFUSCATE_70_80	body
175	1	HTML_OBFUSCATE_90_100	body
176	1.247	HTML_TAG_BALANCE_BODY	body
177	0.520	HTML_TAG_BALANCE_HEAD	body
178	1	HTML_TAG_EXIST_BGSOUND	body
179	1	HTML_BADTAG_40_50	body
180	1	HTML_BADTAG_50_60	body
181	1	HTML_BADTAG_60_70	body
182	1	HTML_BADTAG_90_100	body
183	0.000	HTML_NONELEMENT_30_40	body
184	1	HTML_NONELEMENT_40_50	body
185	1	HTML_NONELEMENT_60_70	body
186	1	HTML_NONELEMENT_80_90	body
187	1	HTML_IFRAME_SRC	body
188	0	NO_DNS_FOR_FROM	header
189	0.406	REMOVE_BEFORE_LINK	body
190	2.699	GUARANTEED_100_PERCENT	body
191	2.683	DEAR_FRIEND	body
192	1.999	DEAR_SOMETHING	body
193	0.001	BILLION_DOLLARS	body
194	2.399	EXCUSE_4	body
195	2.799	EXCUSE_24	body
196	2.907	EXCUSE_REMOVE	body
197	1	STRONG_BUY	body

198	1	STOCK_ALERT	body
199	1	NOT_ADVISOR	body
200	1	PREST_NON_ACCREDITED	body
201	0.927	BODY_ENHANCEMENT	body
202	1.691	BODY_ENHANCEMENT2	body
203	1.539	IMPOTENCE	body
204	3.599	NA_DOLLARS	body
205	2.599	US_DOLLARS_3	body
206	3.799	MILLION_USD	body
207	1.750	URG_BIZ	body
208	2.910	MONEY_BACK	body
209	2.700	FREE_QUOTE_INSTANT	body
210	2.799	BAD_CREDIT	body
211	1	REFINANCE_YOUR_HOME	body
212	1	REFINANCE_NOW	body
213	2.199	NO_MEDICAL	body
214	0.714	DIET_1	body
215	2.699	FIN_FREE	body
216	1	FORWARD_LOOKING	body
217	1.840	ONE_TIME	body
218	0.700	JOIN_MILLIONS	body
219	0.553	MARKETING_PARTNERS	body
220	0.161	LOW_PRICE	body
221	2.699	UNCLAIMED_MONEY	body
222	1	OBSCURED_EMAIL	body
223	1	BANG_OPRAH	body
224	1.404	ACT_NOW_CAPS	body
225	2.799	MORE_SEX	body
226	2.202	BANG_GUAR	body
227	0.200	INVESTMENT_ADVICE	body
228	3.100	MALE_ENHANCE	body
229	0.794	PRICES_ARE_AFFORDABLE	body

230	3.487	REPLICA_WATCH	body
231	0.595	EM_ROLEX	body
232	1	FREE_PORN	body
233	1	CUM_SHOT	body
234	1	LIVE_PORN	body
235	1	SUBJECT_SEXUAL	header
236	1.898	RATWARE_EGROUPS	header
237	1	RATWARE_OE_MALFORMED	header
238	1	RATWARE_MOZ_MALFORMED	header
239	1.153	RATWARE_MPOP_WEBMAIL	header
240	1	RATWARE_HASH_DASH	rawbody
241	1	RATWARE_GECKO_BUILD	header
242	1	X_MESSAGE_INFO	header
243	2.499	HEADER_SPAM	header
244	1	RATWARE_RCVD_PF	header
245	1	RATWARE_RCVD_AT	header
246	2.999	RATWARE_EFROM	header
247	1	HIGH_CODEPAGE_URI	uri
248	0.000	NUMERIC_HTTP_ADDR	uri
249	0.807	HTTP_ESCAPED_HOST	uri
250	0.001	HTTP_EXCESSIVE_ESCAPES	uri
251	0.001	IP_LINK_PLUS	uri
252	0.001	WEIRD_PORT	uri
253	1	YAHOO_RD_REDIR	uri
254	1	YAHOO_DRS_REDIR	uri
255	1	HTTP_77	uri
256	2.999	SPOOF_COM2OTH	uri
257	0.001	SPOOF_COM2COM	uri
258	1	SPOOF_NET2COM	uri
259	2.800	URI_HEX	uri
260	0.500	URI_NOVOWEL	uri
261	1	URI_UNSUBSCRIBE	uri

262	2.299	URI_NO_WWW_INFO_CGI	uri
263	2.399	URI_NO_WWW_BIZ_CGI	uri
264	0.159	NORMAL_HTTP_TO_IP	uri
265	0	BAYES_00	body
266	0	BAYES_05	body
267	0	BAYES_20	body
268	0	BAYES_40	body
269	0	BAYES_50	body
270	0	BAYES_60	body
271	0	BAYES_80	body
272	0	BAYES_95	body
273	0	BAYES_99	body
274	1	ACCESSDB	header
275	0.1	MICROSOFT_EXECUTABLE	body
276	0.1	MIME_SUSPECT_NAME	body
277	0	DCC_CHECK	full
278	0	DCC_REPUT_00_12	full
279	0	DCC_REPUT_13_19	full
280	0	DCC_REPUT_70_89	full
281	0	DCC_REPUT_90_94	full
282	0	DCC_REPUT_95_98	full
283	0	DCC_REPUT_99_100	full
284	0.1	DKIM_SIGNED	full
285	-0.1	DKIM_VALID	full
286	-0.1	DKIM_VALID_AU	full
287	0	DKIM_ADSP_NXDOMAIN	header
288	0	DKIM_ADSP_DISCARD	header
289	0	DKIM_ADSP_ALL	header
290	0.001	DKIM_ADSP_CUSTOM_LOW	header
291	0.001	DKIM_ADSP_CUSTOM_MED	header
292	0.001	DKIM_ADSP_CUSTOM_HIGH	header
293	1	DKIM_VERIFIED	full

294	1	DKIM_POLICY_TESTING	header
295	1	DKIM_POLICY_SIGNSOME	header
296	1	DKIM_POLICY_SIGNALL	header
297	-0.5	HASHCASH_20	header
298	-0.7	HASHCASH_21	header
299	-1.0	HASHCASH_22	header
300	-2.0	HASHCASH_23	header
301	-3.0	HASHCASH_24	header
302	-4.0	HASHCASH_25	header
303	-5.0	HASHCASH_HIGH	header
304	0.1	HASHCASH_2SPEND	header
305	0	PYZOR_CHECK	full
306	0	RAZOR2_CHECK	full
307	0	RAZOR2_CF_RANGE_51_100	full
308	0	RAZOR2_CF_RANGE_E4_51_100	full
309	0	RAZOR2_CF_RANGE_E8_51_100	full
310	1	SUBJECT_FUZZY_MEDS	header
311	0.641	SUBJECT_FUZZY_CHEAP	header
312	1	SUBJECT_FUZZY_PENIS	header
313	1	SUBJECT_FUZZY_TION	header
314	1	FUZZY_AFFORDABLE	body
315	2.199	FUZZY_AMBIEN	body
316	1	FUZZY_BILLION	body
317	0.001	FUZZY_CPILL	body
318	1.699	FUZZY_CREDIT	body
319	2.356	FUZZY_ERECT	body
320	1	FUZZY_GUARANTEE	body
321	1	FUZZY_MEDICATION	body
322	2.599	FUZZY_MILLION	body
323	1	FUZZY_MONEY	body
324	1	FUZZY_MORTGAGE	body
325	1	FUZZY_OBLIGATION	body

326	1	FUZZY_OFFERS	body
327	2.960	FUZZY_PHARMACY	body
328	2.799	FUZZY_PHENT	body
329	1	FUZZY_PRESCRIPT	body
330	1.821	FUZZY_PRICES	body
331	1	FUZZY_REFINANCE	body
332	1	FUZZY_REMOVE	body
333	3.399	FUZZY_ROLEX	body
334	1	FUZZY_SOFTWARE	body
335	1	FUZZY_THOUSANDS	body
336	1	FUZZY_VLIUM	body
337	1	FUZZY_VIOXX	body
338	0.001	FUZZY_VPILL	body
339	2.202	FUZZY_XPILL	body
340	-0.001	SPF_PASS	header
341	0	SPF_NEUTRAL	header
342	0	SPF_FAIL	header
343	0	SPF_SOFTFAIL	header
344	-0.001	SPF_HELO_PASS	header
345	0	SPF_HELO_NEUTRAL	header
346	0	SPF_HELO_FAIL	header
347	0	SPF_HELO_SOFTFAIL	header
348	2.800	UNWANTED_LANGUAGE_BODY	body
349	1.500	BODY_8BITS	body
350	0	URIBL_SBL	body
351	0	URIBL_SC_SURBL	body
352	0	URIBL_WS_SURBL	body
353	0	URIBL_PH_SURBL	body
354	0	URIBL_OB_SURBL	body
355	0	URIBL_AB_SURBL	body
356	0	URIBL_JP_SURBL	body
357	0	URIBL_BLACK	body

358	0	URIBL_GREY	body
359	0.001	URIBL_RED	body
360	1	AWL	header
361	1	SHORTCIRCUIT	header
362	100.000	USER_IN_BLACKLIST	header
363	-100.000	USER_IN_WHITELIST	header
364	-15.000	USER_IN_DEF_WHITELIST	header
365	10.000	USER_IN_BLACKLIST_TO	header
366	-6.000	USER_IN_WHITELIST_TO	header
367	-20.000	USER_IN_MORE_SPAM_TO	header
368	-100.000	USER_IN_ALL_SPAM_TO	header
369	-100.000	USER_IN_DKIM_WHITELIST	header
370	-7.500	USER_IN_DEF_DKIM_WL	header
371	-100.000	USER_IN_SPF_WHITELIST	header
372	-7.500	USER_IN_DEF_SPF_WL	header
373	-100	SUBJECT_IN_WHITELIST	header
374	100	SUBJECT_IN_BLACKLIST	header
375	0.148	APOSTROPHE_FROM	header
376	1	AXB_HELO_HOME_UN	header
377	1	AXB_XMID_1212	header
378	1	AXB_XMID_1510	header
379	1	AXB_XMID_OEGOESNULL	header
380	1	AXB_XM_SENDMAIL_NOT	header
381	1	AXB_XR_STULDAP	header
382	2.399	BANKING_LAWS	body
383	2.370	BASE64_LENGTH_78_79	body
384	1.379	BASE64_LENGTH_79_INF	body
385	1.802	BUG6152_INVALID_DATE_TZ_ABSURD	header
386	0.001	CTYPE_001C_B	header
387	0.001	CURR_PRICE	body
388	1	DEAR_BENEFICIARY	body
389	1	DEAR_EMAIL	body

390	3.099	DEAR_WINNER	body
391	1	DOS_ANAL_SPAM_MAILER	header
392	2.599	DOS_RCVD_IP_TWICE_C	header
393	1	DOS_URI_ASTERISK	uri
394	1	DRUGS_HDIA	header
395	1	FB_ADD_INCHES	body
396	1	FB_ALMOST_SEX	body
397	1	FB_ANA_TRIM	body
398	1	FB_ANUI	body
399	1	FB_BILLI0N	body
400	1	FB_C0MPANY	body
401	1	FB_CAN_LONGER	body
402	1.688	FB_CIALIS_LEO3	body
403	1	FB_DOUBLE_0WORDS	body
404	1	FB_EMAIL_HIER	body
405	0.289	FB_EXTRA_INCHES	body
406	1	FB_FAKE_NUMBERS	body
407	1	FB_FAKE_NUMS4	body
408	1	FB_FHARMACY	body
409	1	FB_FORWARD_LOOK	body
410	1	FB_GAPPY_ADDRESS	body
411	2.314	FB_GET_MEDS	body
412	2.340	FB_GVR	body
413	1	FB_HEY_BRO_COMMA	body
414	1	FB_HG_H_CAP	body
415	1	FB_HOMELOAN	body
416	1	FB_IMPRESS_GIRL	body
417	2.699	FB_INCREASE_YOUR	body
418	2.799	FB_INDEPEND_RWD	body
419	1	FB_L0AN	body
420	1	FB_LETTERS_21B	body
421	0.001	FB_LOSE_WEIGHT_CAP	body

422	1	FB_LOWER_PAYM	body
423	1	FB_MORE_SIZE	body
424	1	FB_NOT_PHONE_NUM1	body
425	1	FB_NOT_PHONE_NUM3	body
426	1	FB_NOT_SCHOOL	body
427	1.656	FB_NO_SCRIP_NEEDED	body
428	1	FB_NUMYO	body
429	1	FB_NUMYO2	body
430	1	FB_ODD_SPACED_MONEY	body
431	1	FB_ONIINE	body
432	1	FB_P1LL	body
433	1	FB_PENIS_GROWTH	body
434	1	FB_PIPEDOLLAR	body
435	1	FB_PIPE_ILLION	body
436	1	FB_PROLONGED_HARD	body
437	3.313	FB_QUALITY_REPLICA	body
438	1	FB_REF_CODE_SPACE	body
439	1.674	FB_REPLICA_ROLEX	body
440	1	FB_REPLIC_CAP	body
441	1	FB_RE_FI	body
442	1	FB_ROLLER_IS_T	body
443	1	FB_ROLX	body
444	2.799	FB_SAVE_PERSC	body
445	2.887	FB_SOFTTABS	body
446	2.499	FB_SPACED_FREE	body
447	0.001	FB_SPACED_PHN_3B	body
448	1	FB_SPACEY_ZIP	body
449	1	FB_SPUR_M	body
450	1	FB_SSEX	body
451	1	FB_STOCK_EXPLODE	body
452	1	FB_SYMBLO	body
453	3.599	FB_THIS_ADVERT	body

454	1	FB_THOUS_PERSONAL	body
455	3.399	FB_TO_STOP_DISTRO	body
456	2.352	FB_ULTRA_ALLURE	body
457	1	FB_UNLOCK_YOUR_G	body
458	1	FB_UNRESOLV_PROV	body
459	1	FB_YOURSELF_MASTER	body
460	1	FB_YOUR_REFI	body
461	1	FH_BAD_OEV1441	header
462	0.000	FH_DATE_IS_19XX	header
463	2.167	FH_FAKE_RCVD_LINE	header
464	4.000	FH_FAKE_RCVD_LINE_B	header
465	1.708	FH_FROMEML_NOTLD	header
466	2.599	FH_FROM_CASH	header
467	2.699	FH_FROM_GET_NAME	header
468	2.599	FH_FROM_GIVEAWAY	header
469	1	FH_FROM_HOODIA	header
470	1.602	FH_HAS_XAIMC	header
471	3.299	FH_HAS_XID	header
472	3.699	FH_HELO_ALMOST_IP	header
473	1	FH_HELO_ENDS_DOT	header
474	1	FH_HELO_EQ_610HEX	header
475	0.607	FH_HELO_EQ_CHARTER	header
476	2.361	FH_HELO_EQ_D_D_D_D	header
477	1	FH_HELO_GMAILSMTP	header
478	2.632	FH_HOST_EQ_DYNAMICIP	header
479	0.001	FH_HOST_EQ_PACBELL_D	header
480	2.681	FH_HOST_EQ_VERIZON_P	header
481	3.199	FH_HOST_IN_ADDRARPA	header
482	1	FH_MSGID_000000	header
483	1	FH_MSGID_01C67	header
484	1	FH_MSGID_01C70XXX	header
485	1	FH_MSGID_REPLACE	header

486	1	FH_MSGID_XXBLAH	header
487	2.399	FH_MSGID_XXX	header
488	1	FH_RE_NEW_DDD	header
489	1	FH_XMAIL_REPLACE	header
490	3.800	FILL_THIS_FORM_LONG	body
491	1	FM_XMAIL_F_OUT	header
492	1	FORGED_RELAY_MUA_TO_MX	header
493	0.001	FRT_ADOBE2	body
494	2.499	FRT_APPROV	body
495	2.523	FRT_BIGGERMEM1	body
496	0.000	FRT_DIPLOMA	body
497	1	FRT_DISCOUNT	body
498	1	FRT_DOLLAR	body
499	1	FRT_ESTABLISH2	body
500	1	FRT_FUCK2	body
501	1	FRT_GUARANTEE1	body
502	1	FRT_INVESTOR	body
503	1	FRT_LEVITRA	body
504	1	FRT_MEETING	body
505	1.681	FRT_OFFER2	body
506	1	FRT_OPPORTUN2	body
507	2.299	FRT_PENIS1	body
508	1	FRT_PHARMAC	body
509	0.001	FRT_PRICE	body
510	1	FRT_REFINANCE1	body
511	2.699	FRT_ROLEX	body
512	1	FRT_SEXUAL	body
513	0.000	FRT_SOMA	body
514	0.001	FRT_SOMA2	body
515	1	FRT_STRONG1	body
516	1	FRT_STRONG2	body
517	1	FRT_SYMBOL	body

518	0.480	FRT_TODAY2	body
519	1	FRT_VALIUM1	body
520	1	FRT_VALIUM2	body
521	1	FRT_WEIGHT2	body
522	1	FRT_XANAX1	body
523	1	FRT_XANAX2	body
524	1	FR_3TAG_3TAG	rawbody
525	2.299	FR_ALMOST_VIAG2	rawbody
526	1	FR_CANTSEETEXT	rawbody
527	1	FR_MIDER	rawbody
528	2.899	FR_TITLE_NUMS	rawbody
529	3.099	FSL_FAKE_GMAIL_RCVD	header
530	2.631	FSL_FAKE_HOTMAIL_RVCD	header
531	2.699	FSL_GEO_ABUSE	uri
532	2.598	FSL_HELO_BARE_IP_1	header
533	1.682	FSL_HELO_DEVICE	header
534	2.361	FSL_HELO_NON_FQDN_1	header
535	1	FSL_HELO_SETUP	header
536	3.899	FSL_INTERIA_ABUSE	uri
537	1	FSL_LSPACES_ABUSE	uri
538	4.199	FSL_YG_ABUSE	uri
539	1.693	FS_ABIGGER	header
540	2.499	FS_APPROVE_YOU	header
541	2.499	FS_AT_NO_COST	header
542	1	FS_CHEAP_CAP	header
543	1	FS_DOLLAR_BONUS	header
544	1	FS_EJACULA	header
545	1	FS_ERECTION	header
546	1	FS_HUGECOCK	header
547	2.645	FS_LARGE_PERCENT2	header
548	1	FS_LOW_RATES	header
549	1	FS_NEW_SOFT_UPLOAD	header

550	1	FS_NEW_XXX	header
551	1	FS_NO_SCRIP	header
552	2.399	FS_NUDE	header
553	2.400	FS_OBFU_PRMCY	header
554	1	FS_PERSCRIPTION	header
555	2.980	FS_PHARMASUB2	header
556	1	FS_RAMROD	header
557	1.630	FS_REPLICA	header
558	3.237	FS_REPLICAWATCH	header
559	1	FS_RE_APPROV	header
560	2.799	FS_START_DOYOU2	header
561	0.249	FS_START_LOSE	header
562	1	FS_TEEN_BAD	header
563	1	FS_TIP_DDD	header
564	1.894	FS_WEIGHT_LOSS	header
565	2.599	FS_WILL_HELP	header
566	1	FS_WITH_SMALL	header
567	1	FUZZY_MERIDIA	body
568	2.801	FU_COMMON_SUBS2	uri
569	1	FU_ENDS_NUMS_DOTS_CLK	uri
570	1	FU_END_ET	uri
571	1	FU_HOODIA	uri
572	1	FU_LONG_QUERY3	uri
573	1	FU_MIDER	uri
574	1	FU_UKGEOCITIES	uri
575	1	FU_URI_TRACKER_T	uri
576	1	GEO_QUERY_STRING	uri
577	1	HDRS_MISSP	header
578	1	HEADER_COUNT_SUBJECT	header
579	1	HELO_FRIEND	header
580	0.001	HELO_LH_HOME	header
581	1	HELO_LH_LD	header

582	2.639	HELO_LOCALHOST	header
583	2.899	HELO_OEM	header
584	4.299	HK_NAME_DRUGS	header
585	1	HK_NAME_FREE	header
586	2.638	HK_RANDOM_ENVFROM	header
587	1	HK_SCAM_N2	body
588	2.762	HS_BOBAX_MID_2	header
589	1	HS_BODY_UPLOADED_SOFTWARE	body
590	0.001	HS_DRUG_DOLLAR_1	body
591	0.001	HS_DRUG_DOLLAR_2	body
592	0.001	HS_DRUG_DOLLAR_3	body
593	1	HS_GETMEOFF	uri
594	1.105	HS_INDEX_PARAM	uri
595	1	HS_MEETUP_FOR_SEX	body
596	1	HS_SUBJ_NEW_SOFTWARE	header
597	1	HS_SUBJ_ONLINE_PHARMACEUTICAL	header
598	3.211	HS_VPXL	body
599	0.557	HTTPS_HTTP_MISMATCH	body
600	1	JM_I_FEEL_LUCKY	uri
601	1	JM_RCVD_QMAILV1	header
602	3.800	KB_DATE_CONTAINS_TAB	header
603	1	KB_RATWARE_OUTLOOK_08	header
604	1	KB_RATWARE_OUTLOOK_12	header
605	1	KB_RATWARE_OUTLOOK_16	header
606	4.400	KB_RATWARE_OUTLOOK_MID	header
607	3.300	LIVEFILESTORE	uri
608	0.001	LONG_TERM_PRICE	body
609	1	LOOPHOLE_1	body
610	1	LOTTO_AGENT	body
611	0.539	L_SPAM_TOOL_13	header
612	1	MID_DEGREES	header
613	1	MIME_BOUND_EQ_REL	header

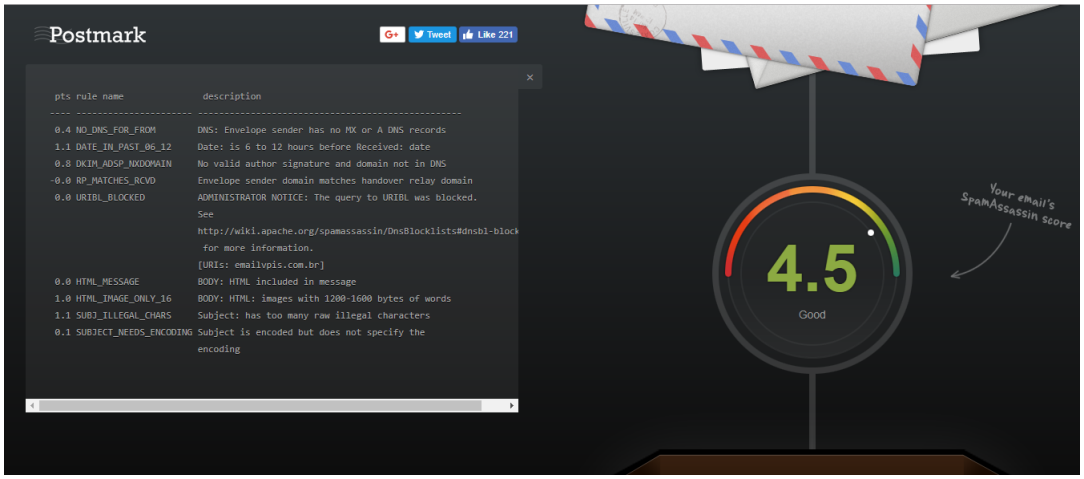
614	0.511	NULL_IN_BODY	full
615	1	RCVD_BAD_ID	header
616	1	RCVD_FORGED_WROTE	header
617	1	RCVD_FORGED_WROTE2	header
618	0	RCVD_IN_BRBL_LASTEXT	header
619	0	RCVD_IN_CSS	header
620	0	RCVD_IN_DNSWL_HI	header
621	0	RCVD_IN_DNSWL_LOW	header
622	0	RCVD_IN_DNSWL_MED	header
623	0	RCVD_IN_DNSWL_NONE	header
624	0	RCVD_IN_IADB_DK	header
625	0	RCVD_IN_IADB_DOPTIN	header
626	1	RCVD_IN_IADB_DOPTIN_GT50	header
627	0	RCVD_IN_IADB_DOPTIN_LT50	header
628	1	RCVD_IN_IADB_EDDB	header
629	1	RCVD_IN_IADB_EPIA	header
630	1	RCVD_IN_IADB_GOODMAIL	header
631	0	RCVD_IN_IADB_LISTED	header
632	1	RCVD_IN_IADB_LOOSE	header
633	1	RCVD_IN_IADB_MI_CPEAR	header
634	1	RCVD_IN_IADB_MI_CPR_30	header
635	0	RCVD_IN_IADB_MI_CPR_MAT	header
636	0	RCVD_IN_IADB_ML_DOPTIN	header
637	1	RCVD_IN_IADB_NOCONTROL	header
638	1	RCVD_IN_IADB_OOO	header
639	0	RCVD_IN_IADB_OPTIN	header
640	0	RCVD_IN_IADB_OPTIN_GT50	header
641	1	RCVD_IN_IADB_OPTIN_LT50	header
642	1	RCVD_IN_IADB_OPTOUTONLY	header
643	0	RCVD_IN_IADB_RDNS	header
644	0	RCVD_IN_IADB_SENDERID	header
645	0	RCVD_IN_IADB_SPF	header

646	1	RCVD_IN_IADB_UNVERIFIED_1	header
647	1	RCVD_IN_IADB_UNVERIFIED_2	header
648	1	RCVD_IN_IADB_UT_CPEAR	header
649	1	RCVD_IN_IADB_UT_CPR_30	header
650	0	RCVD_IN_IADB_UT_CPR_MAT	header
651	0	RCVD_IN_PSBL	header
652	0.0	RCVD_IN_RP_CERTIFIED	header
653	0	RCVD_IN_RP_RNBL	header
654	0.0	RCVD_IN_RP_SAFE	header
655	1	RCVD_MAIL_COM	header
656	3.700	RDNS_LOCALHOST	header
657	1.712	SANE_04e8bf28eb445199a7f11b943c44d209	body
658	3.199	SANE_1c4f3286fa4aed6424ced88bfaf8b09c	body
659	2.976	SANE_2b173a7fb7518c75ac8a2d294d773fd8	body
660	0.001	SANE_3b92eda751c992f230f215fb7eb36844	body
661	2.231	SANE_4ef8302546bf270a19baf98508afacc4	body
662	3.999	SANE_7429530a7398f43f1f1b795f9420714e	body
663	3.099	SANE_91eb43f705d25c804374a746d7519660	body
664	3.799	SANE_d0d2b0f6373bf91253d66dd74c594b87	body
665	0.001	SHORT_TERM_PRICE	body
666	1.898	STOX_REPLY_TYPE	header
667	1	TAB_IN_FROM	header
668	2.599	THEBAT_UNREG	header
669	0.748	TT_MSGID_TRUNC	header
670	1	TVD_ACT_193	body
671	2.356	TVD_APPROVED	body
672	1	TVD_DEAR_HOMEOWNER	body
673	1	TVD_ENVFROM_APOST	header
674	0.001	TVD_FINGER_02	header
675	1	TVD_FLOAT_GENERAL	rawbody
676	1	TVD_FUZZY_DEGREE	body
677	1	TVD_FUZZY_FINANCE	body

678	1	TVD_FUZZY_FIXED_RATE	body
679	1	TVD_FUZZY_MICROCAP	body
680	1	TVD_FUZZY_PHARMACEUTICAL	body
681	1	TVD_FUZZY_SYMBOL	body
682	1.529	TVD_INCREASE_SIZE	body
683	1	TVD_LINK_SAVE	body
684	1	TVD_PCT_OFF	header
685	1.201	TVD_PH_BODY_ACCOUNTS_PRE	body
686	3.127	TVD_PH_REC	body
687	0.291	TVD_PH_SEC	body
688	2.602	TVD_PH_SUBJ_ACCOUNTS_POST	header
689	2.284	TVD_PH_SUBJ_SEC_MEASURES	header
690	1.251	TVD_PH_SUBJ_URGENT	header
691	2.697	TVD_QUAL_MEDS	body
692	1	TVD_RATWARE_CB	header
693	1	TVD_RATWARE_CB_2	header
694	1	TVD_RATWARE_MSGID_02	header
695	0.001	TVD_RCVD_IP	header
696	0.159	TVD_RCVD_IP4	header
697	0.242	TVD_RCVD_SINGLE	header
698	0.001	TVD_RCVD_SPACE_BRACKET	header
699	1	TVD_SECTION	body
700	1	TVD_SILLY_URI_OBFU	body
701	1	TVD_SPACED_SUBJECT_WORD3	header
702	1	TVD_STOCK1	body
703	0.001	TVD_SUBJ_ACC_NUM	header
704	1	TVD_SUBJ_FINGER_03	header
705	1	TVD_SUBJ_OWE	header
706	2.599	TVD_SUBJ_WIPE_DEBT	header
707	1.957	TVD_VISIT_PHARMA	body
708	1	TVD_VIS_HIDDEN	rawbody
709	0	URIBL_RHS_DOB	body

710	3.099	URI_OBFU_WWW	body
711	2.886	X_MAILER_CME_6543_MSN	header

Figura 2 – Teste de *email* utilizando a *API* com o *SpamAssassin*



Fonte: Elaborado pelo autor

Figura 3 – Descrição de testes realizados pelo *SpamAssassin*

AREA TESTED	LOCALE	DESCRIPTION OF TEST	TEST NAME	DEFAULT SCORES *(local, net, with bayes, with bayes+net)*	MORE INFO *(additional wiki docs)*
body		Generic Test for Unsolicited Bulk Email	GTUBE	1000.000	
full		Listed in Razor2 (http://razor.sf.net/)	RAZOR2_CHECK	0 0.150 0 1.511	
body		Razor2 gives confidence level above 50%	RAZOR2_CF_RANGE_51_100	0 1.485 0 0.056	
full		Listed in DCC (http://rhyolite.com/anti-spam/dcc/)	DCC_CHECK	0 1.373 0 2.169	
full		Listed in Pyzor (http://pyzor.sf.net/)	PYZOR_CHECK	0 2.041 0 3.451	
body		Incorporates a tracking ID number	TRACKER_ID	1.825 1.064 1.818 0.555	
body		Weird repeated double-quotation marks	WEIRD_QUOTING	1.353 1.966 1.774 2.000	
rawbody		Extra blank lines in base64 encoding	MIME_BASE64_BLANKS	0.693 0.819 1.391 1.469	
rawbody		base64 attachment does not have a file name	MIME_BASE64_NO_NAME	0.022 0 0.017 0.000	
rawbody		Message text disguised using base64 encoding	MIME_BASE64_TEXT	1.780 0.110 1.403 0.298	
rawbody		MIME section missing boundary	MIME_MISSING_BOUNDARY	0 0.247 0.224 0	
body		Multipart message mostly text/html MIME	MIME_HTML_MOSTLY	1.540 0.285 0.713 1.023	
body		Message only has text/html MIME parts	MIME_HTML_ONLY	1.204 1.158 1.156 0.177	
rawbody		Quoted-printable line longer than 76 chars	MIME_QP_LONG_LINE	0 0.000 0.105 0.039	
rawbody		MIME filename does not match content	MIME_SUSPECT_NAME	0.100	
body		HTML and text parts are different	MPART_ALT_DIFF	1.837 1.505 1.823 0.066	
body		Character set indicates a foreign language	CHARSET_FARAWAY	3.200	
body		Message written in an undesired language	UNWANTED_LANGUAGE_BODY	2.800	
body		Body includes 8 consecutive 8-bit characters	BODY_8BITS	1.500	
body		Body contains a ROT13-encoded email address	EMAIL_ROT13	2.720 1.474 2.934 3.105	
body		Message body has 70-80% blank lines	BLANK_LINES_70_80	1.668 1.127 0.745 1.515	
body		Message body has 80-90% blank lines	BLANK_LINES_80_90	0.046 0 0.216 0	
body		Message body has 90-100% blank lines	BLANK_LINES_90_100	1.490 1.750 1.877 1.996	
body		Message body has many words used only once	UNIQUE_WORDS	3.109 2.549 1.639 2.273	

Fonte: Elaborado pelo autor