Impacto da Evolução Temporal na Detecção de Spammers na Rede de Origem*

Pedro Henrique B. Las Casas[1], Jussara M. Almeida[1], Marcos A. Gonçalves[1], Dorgival Guedes[1], Artur Ziviani,[2] Humberto T. Marques-Neto[3]

[1] Departamento de Ciência da Computação, Universidade Federal de Minas Gerais
[2]Coordenação de Ciência da Computação, Laboratório Nacional de Computação Científica
[3]Departamento de Ciência da Computação, Pontifícia Universidade Católica de Minas Gerais

{pedro.lascasas,jussara,mgoncalv,dorgival}@dcc.ufmg.br

ziviani@lncc.br, humberto@pucminas.br

Abstract. *The large volume of unwanted email traffic (spam) circulating on the Internet consumes resources that could be better used. In a previous work, we proposed the method SpaDeS for spammers detection in the source network, which uses only the network layer metrics. In this paper, we present a more detailed analysis of Spades, focusing particularly on its sensitivity to temporal aspects. We also propose the use of a strategy that uses active learning to select new training examples to mitigate the problem of loss of efficacy over time. The method was applied to a real dataset and the results show that despite some variation in performance, it maintains the effectiveness fairly stable over a month. The use of active learning to select the training set results in performance gain of about 8% in classifying legitimate users with little loss in classification of spammers (less than 3%).*

Resumo. *O grande volume de tráfego de e-mails indesejados (spam) que circulam na Internet consome recursos que poderiam ser melhor utilizados. Em um trabalho anterior, propusemos o método SpaDeS de detecção de spammers na rede de origem, que utiliza apenas métricas da camada de rede. Neste trabalho, apresentamos uma análise mais detalhada do SpaDes, focando particularmente na sua sensibilidade a aspectos temporais. Propomos também o uso de uma estratégia que utiliza aprendizado ativo para seleção de novos exemplos de treino de forma a mitigar o problema de perda de eficácia ao longo do tempo. O método foi novamente aplicado a um conjunto de dados reais e os resultados obtidos mostram que, apesar de certa variação no desempenho, ele mantém a eficácia razoavelmente estável ao longo de um mês. O uso de aprendizado ativo para seleção do conjunto de treino resulta em ganho de desempenho de cerca de 8% ao classificar usuários legítimos com poucas perda na classificação dos spammers (menos de 3%).*

1. Introdução

Apesar do alto número de técnicas anti-spam disponíveis atualmente [Schryen 2007], sendo algumas largamente adotadas pelos prestadores de

*Este trabalho é parcialmente financiada pelo Instituto Nacional de Ciência e Tecnologia para a Web - INCTWeb (MCT/CNPq 573871/2008-6), CNPq, Capes, FAPEMIG, FAPERJ e INWeb.

serviço de correio eletrônico visando reduzir a quantidade de spam nas caixas de entrada dos usuários, *spam* permanece sendo uma grande parte do tráfego na Internet. De fato, nos últimos anos, quase 80% de todos os e-mails que trafegam na rede é *spam* [Fletcher 2009], sendo muitas vezes relacionados com a propagação de *malwares*, como cavalos de tróia, vírus e *worms* [Newman et al. 2002]. O volume de spam na rede permanece alto porque, apesar de eficazes, técnicas *anti-spam* são normalmente empregadas no servidor de correio eletrônico de destino (ou em um intermediário adequado). Assim, elas agem, na melhor das hipóteses, somente após as mensagens indesejadas terem percorrido (pelo menos parcialmente) o caminho de rede entre sua fonte e o ponto de filtragem. Como consequência, tais técnicas não evitam o desperdício de recursos de rede. Além disso, a execução do próprio filtro no servidor de destino consome recursos de processamento e de memória local.

Em um trabalho anterior [Las-Casas et al. 2011] propusemos um método a ser utilizado como complemento às técnicas *anti-spam* presentes no servidor receptor. O método proposto, SpaDeS — *Spammer Detection at the Source*, é utilizado para detecção de *spammers* ainda na rede de origem dos remetentes. SpaDeS explora apenas métricas que não requerem inspeção do conteúdo das mensagens, podendo ser aplicado, por exemplo, em provedores de acesso à Internet de banda larga. Através da análise do tráfego SMTP (*Simple Mail Transfer Protocol*), o SpaDeS é capaz de detectar a ação de possíveis *spammers*. De posse dessa informação, o administrador da rede pode adotar medidas de acordo com a política do provedor, como o bloqueio do tráfego daquela origem, ou outras medidas menos drásticas que não prejudiquem usuários legítimos que porventura gerem falsos-positivos, como o envio de mensagens de alerta para o usuário, uso (periódico) de desafios para testar a legitimidade dos usuários suspeitos, ou a introdução de atrasos nas mensagens daqueles usuários.

O método SpaDeS baseia-se em uma técnica de classificação supervisionada, o que significa que ele requer um conjunto de treino consistente, formado por exemplos de usuários previamente classificados em cada classe de interesse (p.ex: *spammers* e usuários legítimos) por meio de informações externas. O algoritmo de classificação supervisionada constrói um modelo de classificação que permite diferenciar usuários legítimos de *spammers* com base em padrões inferidos do conjunto de treino. Como as características de usuários legítimos e de *spammers* estão sujeitas a sofrerem alterações ao longo do tempo, faz-se necessário a reconstrução do modelo de classificação periodicamente, visando capturar possíveis mudanças no comportamento dos usuários. Para tal, são realizados retreinamentos periódicos para reaprender as características dos usuários e manter a efetividade do método. Como forma de reduzir a frequência com que dados externos sejam necessários para construir uma nova base de treinamento, o SpaDeS baseia-se em uma estratégia iterativa. Essa estratégia faz uso do resultado da classificação anterior, selecionando os usuários cujas classificações apresentem maior confiança para comporem o conjunto de treino a ser utilizado na classificação seguinte [Las-Casas et al. 2011].

Apesar de automática, a estratégia iterativa proposta pode levar à introdução de erros no conjunto de treino, dado que usuários selecionados para compor o treino da próxima iteração podem ter sido classificados erroneamente, apesar das confianças atribuídas as suas classificações serem altas. A introdução de erros no treino pode afetar significativamente a eficácia do método à medida que novas iterações são executadas. Assim, o

objetivo principal deste trabalho é analisar o impacto causado pela estratégia iterativa na eficácia do SpaDeS, considerando sua sensibilidade à evolução temporal, um ponto chave de seu funcionamento que não foi avaliado em [Las-Casas et al. 2011]. Para isso, utiliza-se um conjunto de dados reais, agregados por dia, contendo informações anonimizadas de transações SMTP de usuários de um provedor brasileiro de Internet de banda larga residencial coletadas em 2010. Os experimentos foram realizados nos 28 dias que compõem a base: o SpaDeS foi treinado com os dados coletados no primeiro dia e aplicado, utilizando a estratégia iterativa proposta, nos 27 dias seguintes. Os resultados obtidos indicaram que, apesar de certa variação no desempenho, o SpaDeS tende a permanecer com boa eficácia durante todo o período, classificando corretamente pelo menos 75% e 82% dos usuários legítimos e *spammers*, respectivamente, observados em cada dia.

Apesar da razoável eficácia com apenas um treinamento com dados externos durante 28 dias (período total dos dados), investiga-se ainda estratégias para mitigar a possível perda de eficácia da aplicação sucessiva do SpaDeS, particularmente quando aplicado durante um período mais longo. Para tal, propõe-se uma nova abordagem de seleção do conjunto de treino, a ser utilizada juntamente da proposta anterior [Las-Casas et al. 2011]. Essa abordagem utiliza uma técnica de amostragem ativa, *Active Lazy Associative Classifier* (ALAC) [Silva et al. 2011, Benevenuto et al. 2012], que seleciona os usuários mais informativos de cada classe para comporem o treino. Usuários mais informativos apresentam comportamentos mais diverso em relação aos demais e, portanto, auxiliam na descoberta de padrões e na criação de um modelo mais robusto. A seleção de usuários com base na capacidade informativa permite reduzir bastante a necessidade de treino, quando comparado a técnicas supervisionadas tradicionais, mantendo, muitas vezes, a mesma eficácia. Em outras palavras, o ALAC tende a selecionar uma fração muita pequena de usuários. Estes usuários devem ser rotulados manualmente, ou seja, devem ter suas classes determinadas pelo administrador do SpaDeS (ou por outras fontes).

Para avaliar a nova abordagem proposta, primeiramente opta-se por verificar o potencial de uso do ALAC na geração de treino inicial para o SpaDeS. Para tal, o ALAC foi aplicado em uma base de dados do mesmo provedor, coletada em 2009 para selecionar um conjunto de treino. Esse conjunto, composto de 129 usuários, foi usado como entrada para a classificação, usando o SpaDeS, da base completa de 2010. Os resultados indicaram a classificação correta de mais de 99% dos usuários legítimos e cerca de 84% dos *spammers*, um desempenho muito bom, comparável ao método original.

Em seguida, verifica-se o benefício de usar o ALAC em conjunto com a estratégia iterativa proposta [Las-Casas et al. 2011]. Em outras palavras, repetiu-se a avaliação do SpaDeS ao longo dos 28 dias que compõem a base de 2010, utilizando os usuários classificados com maior confiança na iteração (dia) anterior bem como novos usuários selecionados pelo ALAC (e pré-classificados manualmente) para comporem o conjunto de treino para a próxima iteração. A nova abordagem levou a uma melhora de cerca de 8% para classificação de usuários legítimos com uma pequena perda de desempenho de menos de 3% na classificação de *spammers*. Mais importante, os resultados indicam uma flexibilidade na escolha de diferentes estratégias de acordo com o cenário trabalhado, dependendo do compromisso entre a classificação correta de usuários legítimos e *spammers*.

Em suma, as principais contribuições deste trabalho são: (i) a proposta de uma estratégia de aprendizado ativo como forma de mitigar a esperada perda de eficácia do

método e como alternativa para a melhoria do processo como um todo; e (ii) a avaliação da sensibilidade do SpaDeS à evolução temporal.

A seguir, a Seção 2 discute trabalhos relacionados. A Seção 3 apresenta o método de detecção de *spammers* proposto bem como a nova estratégia proposta. A Seção 4 apresenta a base de dados utilizada e a Seção 5 discute os resultados mais relevantes. Conclusões e trabalhos futuros são discutidos na Seção 6.

2. Trabalhos Relacionados

Para o desenvolvimento de métodos para a detecção de *spammers*, é necessário o entendimento das características de *spams*, uma vez que as peculiaridades apresentadas por estas mensagens podem ser fundamentais na diferenciação daqueles que as enviam. Kim *et al.* caracterizaram o tráfego de *spams* a partir de dados da camada de aplicação coletados no destino da mensagem e mostraram que o intervalo entre chegadas de *spams* é bem inferior ao intervalo entre e-mails legítimos (menor que 5 segundos em 95% dos casos) [Kim e Choi 2008]. Gomes *et al.* analisaram uma carga de trabalho de mensagens de usuários de uma universidade brasileira e destacaram uma série de características capazes de diferenciar *spams* de mensagens legítimas [Gomes et al. 2007]. Em uma extensão daquele trabalho, os mesmos autores indicaram que o tráfego legítimo apresenta menor entropia que o tráfego gerado pelos *spammers*, os quais, geralmente, enviam e-mails indistintamente para os seus alvos [Gomes et al. 2009].

Utilizando propriedades do tráfego de rede para determinar se uma mensagem é *spam*, Ouyang *et al.* mostraram recentemente que apenas métricas de um único pacote ou apenas métricas de fluxo não são eficientes para classificação de *spams* por si só, mas que a combinação dos conjuntos de métricas aumenta a eficácia da classificação [Ouyang et al. 2011]. Clayton *et al.* propuseram um projeto chamado *SpamHINTS*, que visa desenvolver técnicas para, através da análise de pacotes do protocolo SMTP, inferir padrões indicativos de atividades de *spamming* [Richard Clayton 2006]. Venkataraman *et al.* estudaram a eficácia de utilizar o comportamento histórico de endereços de IP para prever se um e-mail é legítimo ou *spam* [Venkataraman et al. 2007].

Com relação à análise de remetentes de e-mails, Duan *et al.* analisaram características do comportamento de *spammers* que são críticas para controle de *spam*. Dentre os resultados obtidos, os autores mostraram que a maioria dos servidores de e-mail tendem a enviar somente *spams* ou apenas mensagens legítimas [Duan et al. 2011]. Xie *et al.* mostraram que a grande maioria de servidores de e-mail executando sobre endereços IP dinâmicos são utilizados somente para envio de *spams* [Xie et al. 2007], enquanto que Guerra *et al.* analisaram os padrões de comunicação presentes em uma campanha de *spam* [Guerra et al. 2009].

Diversos trabalhos propuseram soluções para identificação de *spammers* em pontos intermediários da rede. Ramachandran e Feamster investigaram características de tráfego coletado da camada de rede, tais como a persistência de endereços IP e de rotas e características específicas de *botnets*, que sejam comuns a *spammers* [Ramachandran e Feamster 2006]. Já Hao *et al.* aplicaram técnicas de aprendizado de máquina em dados coletados da camada de rede para classificar *usuários* em legítimos e *spammers* em um servidor posicionado entre as redes de origem e destino [Hao et al. 2009]. Schatzmann *et al.* propuseram a detecção de *spammers* no nível

de sistemas autônomos (AS), coletando e combinando as visões locais de múltiplos servidores de e-mail destinatários [Schatzmann et al. 2009]. Ao contrário desses trabalhos, anteriormente [Las-Casas et al. 2011] propôs-se a detecção dos *spammers* ainda na rede de origem, para minimizar o desperdício de recursos devidos ao processo de recepção das mensagens. Aquele trabalho apresenta apenas uma avaliação inicial do método proposto — SpaDeS, deixando em aberto, por exemplo, o impacto da evolução temporal na eficácia do método. Este trabalho complementa o anterior, estendendo a avalição de SpaDeS para considerar a sua aplicação sucessiva em um período de um mês, além de avaliar estratégias alternativas para criação dos conjuntos de treino.

Através da análise de características de fluxos de pacotes SMTP, Sperotto *et al.* propuseram um algoritmo para detecção de *spams* utilizando apenas informações da camada de rede (p.ex: tempo de inatividade e quantidade de picos no fluxos de pacotes) [Sperotto et al. 2009]. Taveira e Duarte propuseram um mecanismo *anti-spam* baseado em autenticação e reputação dos usuários objetivando minimizar falsos positivos ao classificar *spams* [Taveira e Duarte 2008].

Por outro lado, outros métodos de detecção de *spams* utilizaram técnicas de classificação supervisionada, porém exploraram características do conteúdo das mensagens [Kolcz e Alspector 2001, Lakshmi e Radha 2010]. O SpaDeS [Las-Casas et al. 2011] explora técnicas semelhantes, mas considera apenas métricas relacionadas aos protocolos envolvidos, sem inspecionar o conteúdo das mensagens, para garantir a privacidade dos usuários legítimos, e tem como alvo a detecção de *spammers*.

Por fim, o método ALAC de amostragem ativa para classificação supervisionada já foi usado na detecção de poluidores de conteúdo no YouTube, para reduzir do custo de criação do conjunto de treino [Benevenuto et al. 2012]. O seu uso combinado ao SpaDeS visando mitigar o impacto da evolução temporal é uma contribuição deste trabalho.

3. SpaDeS e Sua Estratégia Iterativa

O método SpaDeS (*Spammer Detection at the Source*) [Las-Casas et al. 2011] utiliza um algoritmo de classificação supervisionada que "aprende" um modelo de classificação de usuários a partir de um conjunto de exemplos previamente rotulados (conjunto de treino). O classificador recebe como entrada o número de classes distintas C e exemplos de usuários de cada uma. Após a fase de aprendizado, o modelo derivado pode então ser aplicado para classificar novos usuários (conjunto de teste) nas classes pré-definidas. O método tem uma abordagem iterativa, selecionando novos usuários de treino de maneira automatizada a cada iteração para realizar a classificação seguinte.

Em seguida, a Seção 3.1 apresenta o modelo de representação dos usuários. O algoritmo de classificação utilizado é apresentado na Seção 3.2 e a Seção 3.3 discute as estratégias originalmente propostas para seleção do conjunto de treino. A nova estratégia baseada em amostragem ativa é apresentada na Seção 3.4.

3.1. Modelo de Representação de Usuários

Cada usuário é representado por um vetor de N atributos que conjuntamente descrevem seu comportamento quanto ao uso do protocolo SMTP. Para detectar *spammers* na rede de origem com eficiência, foram utilizados N=5 atributos, que não envolvem

processamento do corpo da mensagem. As métricas são: número de transações SMTP realizadas, número de servidores SMTP de destino distintos acessados, tamanho médio das transações SMTP, distância geodésica[1] média entre origem e destino e tempo médio entre transações consecutivas (aqui referenciado como IATs, *inter-arrival times*). A escolha das métricas foi inspirada nos resultados de um trabalho anterior [Castilho et al. 2010], que caracteriza o tráfego SMTP mostrando que tais métricas são eficazes para diferenciação dos usuários que fazem uso deste tráfego.

3.2. Algoritmo de Classificação Supervisionada

O algoritmo de classificação utilizado é o *Lazy Associative Classifier* (LAC) [Veloso et al. 2006], que tem ótima escalabilidade, com complexidade de tempo polinomial. Diferentemente de outros classificadores, o LAC fornece uma estimativa da confiança na predição feita para cada usuário. Essa confiança pode ser interpretada como uma probabilidade de acerto da classificação. O LAC explora o fato de que, frequentemente, há fortes associações entre os valores dos atributos e as classes. Tais associações estão geralmente implícitas no conjunto de treino e, quando descobertas, revelam aspectos que podem ser utilizados para prever as classes dos usuários. O LAC produz um modelo de classificação composto de regras $\mathcal{X} \rightarrow c_i$ que indicam a associação entre um conjunto de valores de atributos $\mathcal{X}$ e uma classe c_i. O LAC "aprende" esse modelo em duas etapas: (1) extração de regras e (2) predição das classes.

A extração de regras de associação do conjunto de treino é feita sob demanda, com base nos usuários do conjunto de teste. Em outras palavras, para cada usuário u no *conjunto de teste*, ele projeta e filtra o conjunto de treino de acordo com os valores dos atributos de u, extraindo regras desse conjunto filtrado. Assim, ele garante que somente regras com informação relevante para u sejam extraídas do conjunto de treino, reduzindo o número de possíveis regras. O LAC então estima uma confiança $\theta(\mathcal{X} \rightarrow c_i)$ para cada regra $\mathcal{X} \rightarrow c_i$ extraída. Considerando que um usuário u contém todos os valores de atributos contidos em $\mathcal{X}$, $\theta(\mathcal{X} \rightarrow c_i)$ estima a probabilidade condicional da classe de u ser c_i, com base nos atributos contidos em $\mathcal{X}$. Para prever a classe de u, o LAC combina todas as regras $\mathcal{X} \rightarrow c_i$ onde $\mathcal{X}$ contém valores de atributos que coincidem com os de u. Cada regra é tratada como um voto para que a classe de u seja c_i. A probabilidade da classe de u ser c_i é estimada pela confiança média de todos os votos para c_i. Considera-se como a classe de u aquela com maior probabilidade.

Dois parâmetros principais do LAC são o tamanho máximo das regras (número de atributos em $\mathcal{X}$) e a confiança mínima permitida. Considera-se o tamanho máximo como 5 e uma confiança mínima de 0,01, valores de referência comumente usados.

3.3. Operação Iterativa

O funcionamento de qualquer método de classificação supervisionada depende primariamente de um conjunto de treino contendo usuários pré-classificados. A obtenção desse conjunto para a classificação de usuários em *spammers* e legítimos é um grande desafio, uma vez que tais dados tipicamente não estão disponíveis publicamente. Um fator complicador é que almeja-se detectar *spammers* ainda na rede de origem. Logo, faz-se necessário um conjunto de treino coletado naquele ponto do sistema. Caso contrário,

[1]Menor distância entre dois pontos ao longo da superfície da Terra.

os padrões levantados poderiam não generalizar para o conjunto de teste, resultando em um desempenho limitado do classificador.

Uma estratégia proposta anteriormente [Las-Casas et al. 2011] para seleção do treino inicial a ser utilizado pelo SpaDeS consiste na execução do algoritmo não-supervisionado de agrupamento *X-Means* [Pelleg e Moore 2000] sobre dados coletados previamente a fim de identificar perfis (ou classes) de usuários. Para cada classe identificada, seleciona-se os M usuários mais próximos do centróide do grupo correspondente, visando obter bons representantes de cada classe. Utiliza-se como a classe do usuário o grupo identificado pelo algoritmo. Para as classes de usuários com padrão abusivo, indicativo de possível atividade de *spamming*, optou-se naquele trabalho por, ao invés da seleção de usuários próximos do centróide, utilizar dados de uma fonte externa potencialmente mais confiável, uma vez que os mesmos estavam disponíveis. Especificamente, utilizou-se como usuários representativos de *spammers* aqueles cujas máquinas foram apontadas como origem de *spam* por relatos oriundos de outros provedores. Tais relatos, enviados para o endereço `abuse` do provedor que forneceu os dados utilizados nesse trabalho, são gerados por provedores tanto a partir de reclamações de seus usuários (como o recurso "*Report spam*" do Gmail) ou por mecanismos automáticos, como listas de bloqueio ou outros mecanismos de detecção automática de *spam*. Porém, aponta-se que, na ausência desses relatos, a escolha com base no centróide de cada grupo deve ser aplicada.

Visando reduzir a frequência de uso da estratégia baseada em dados do `abuse`, que está sujeita a um bom agrupamento dos usuários e a dados de fontes externas que nem sempre estão disponíveis, propõe-se também uma estratégia iterativa que utiliza os usuários previamente classificados com maior confiança como conjunto de treino para a classificação seguinte. Essa estratégia considera sucessivos conjuntos de teste $t_1, t_2 \cdots t_n$ e seleciona como treino para a classificação do teste t_i, os usuários do conjunto t_{i-1} que foram classificados com uma confiança superior a um certo limiar. O algoritmo 1 apresenta a estratégia utilizada. Ele garante que pelo menos α% dos usuários de cada classe sejam selecionados, mantendo uma confiança mínima uniforme entre todas as classes.

Algoritmo 1: Dados os usuários classificados pelo LAC na iteração anterior, faça:
1. Ordene os usuários de cada classe em ordem decrescente de confiança;
2. Selecione α% dos usuários de cada classe, ordenados anteriormente;
3. Seja c_i^{min} a menor confiança dos usuários selecionados da classe i ($i = 1..n$);
4. Seja $c = min(c_1^{min}, ..., c_n^{min})$;
5. Selecione para o conjunto de treino todos os usuários que possuem confiança $\geq c$, mantendo, para cada um, a classe definida pelo LAC na iteração anterior.

3.4. Amostragem Ativa para Seleção do Conjunto de Treino

A solução iterativa proposta reduz a necessidade de bons agrupamentos e dados de fontes externas (p.ex: `abuse`) mas tem o potencial de introduzir erros no conjunto de treino, uma vez que o LAC não está livre de atribuir uma confiança alta a um usuário classificado erroneamente. A introdução de erros no treino pode afetar significativamente a eficácia da classificação. Além disso, a solução iterativa pode não se adaptar bem a mudanças nos padrões de comportamento dos usuários. Visando minimizar a potencial perda de desempenho dessa solução ao longo do tempo, propõe-se utilizar, em conjunto

com ela, um método de seleção ativa do conjunto de treino denominado ALAC — *Active Lazy Associative Classifier* [Silva et al. 2011].

O ALAC tende a selecionar um conjunto bem reduzido de treino, sem entretanto afetar a eficácia da classificação [Benevenuto et al. 2012]. Esses usuários devem ser pre-classificados manualmente, o que incorre em um custo adicional para o administrador do sistema. Entretanto, como o número de usuários selecionados tende a ser pequeno, esse custo é baixo e pode ser viável e compensatório. Por exemplo, a classificação manual permite ao sistema se adaptar mais rapidamente a mudanças nos padrões de comportamento dos usuários. De forma a reduzir o custo dessa classificação manual ao máximo, propõe-se construir um conjunto de treino a cada nova iteração aplicando inicialmente a estratégia original de selecionar os usuários previamente classificados com maior confiança e em seguida o ALAC. Apenas usuários novos selecionados pelo ALAC devem ser classificados manualmente e inseridos no treino.

Na seleção ativa, deseja-se selecionar, a partir de um conjunto de usuários não classificados, um subconjunto pequeno de instâncias (usuários) que seja mais útil e eficaz para o processo de classificação supervisionada. Esses usuários selecionados serão então classificados manualmente pelo administrador. Adicionalmente, esse conjunto reduzido que foi selecionado pode melhorar a eficácia da classificação pois pode conter menos "ruído" do que o conjunto selecionado pelo SpaDeS. O método proposto de seleção (ALAC) funciona como descrito a seguir. O primeiro usuário selecionado é aquele que tem mais valores de atributos em comum com todos os demais usuários do conjunto. Esse usuário é o mais "representativo" do conjunto. A partir daí, o processo de geração de regras do LAC é utilizado para avaliar cada usuário do conjunto de entrada: aquele usuário que, ao ser avaliado no contexto do conjunto já classificado, gera a menor quantidade de regras, é selecionado para classificação manual. A intuição por trás desse processo é que o usuário que gera menos regras é aquele mais "distinto" dos demais já classificados manualmente e, portanto, o que trará mais informação para a classificação. Ou seja, ao selecionar um usuário cujos valores de atributos derivam poucas regras, estamos aumentando a "diversidade" do nosso treino (do conjunto de exemplos já selecionados e classificados manualmente) e, portanto, aumentando sua capacidade de generalização, isto é, de gerar regras eficazes para usuários ainda não vistos (conjunto de teste). Esse processo de seleção é realizado várias vezes, até que um usuário já selecionado seja escolhido novamente. Nesse momento, o algoritmo converge, tendo selecionado do conjunto de entrada todas as instâncias interessantes.

4. Bases de Dados

Este trabalho utiliza quatro bases de dados diferentes, sendo que duas delas refletem o tráfego SMTP de um provedor de Internet de banda larga e duas contêm listas de usuários daquele provedor que foram denunciados como *spammers* através do endereço `abuse` daquele provedor durante o período considerado.

Cada base contém um *log* de tráfego e um *log* do serviço DHCP do provedor, ambos cobrindo um mesmo período. Os *logs* de tráfego foram coletados por equipamentos da plataforma *Cisco Service Control Engine* (SCE) [Cisco 2010] e contêm amostras do uso da infra-estrutura do provedor de acesso. Os *logs* de tráfego são formados por *transações*. Cada transação representa uma conexão TCP ou um fluxo de dados UDP,

contendo informações como endereços IP de origem e de destino, serviço/protocolo utilizado, data/hora inicial, duração e volume de bytes enviados e recebidos. Os *logs* do serviço DHCP permitem associar transações a usuários através do mapeamento dos endereços físicos de suas máquinas (*MAC addresses*) para os endereços IP fornecidos pelo provedor, com base na data e hora presentes nos dois *logs*. Vale ressaltar que os dados dos usuários foram anonimizados, por questões de segurança e privacidade, removendo todos os campos contendo informações dos usuários.

As bases cobrem os períodos de 01 a 28 de março de 2009 e 12 de junho a 09 de julho de 2010. Cada base passou por um processo de filtragem, sendo removidas transações: (1) que não usavam SMTP; (2) com duração, número de bytes enviados e/ou recebidos iguais a zero, consideradas erros de coleta; (3) que enviaram menos de 160 bytes ou receberam menos de 80 bytes. Estes últimos limiares foram definidos por corresponderem ao número mínimo de bytes para se estabelecer e encerrar uma conexão TCP, considerando 40 bytes para os cabeçalhos IP e TCP nos pacotes de *three-way handshake* e de finalização. Após filtragem, restaram 6,3 milhões de transações associadas a 5.479 usuários na base de 2009, e 5 milhões de transações associadas a 5.389 usuários na base de 2010.

As duas outras bases de dados contêm denúncias recebidas pelo endereço `abuse` do provedor durante os períodos das bases de tráfego, identificando certos usuários como *spammers*. Os e-mails de denúncia informam o endereço IP de origem do *spam* e a data/hora do seu recebimento e estão no formato ARF (*Abuse Reporting Format*), utilizado para mensagens desse tipo [Shafranovich et al. 2010]. Foi desenvolvida uma ferramenta de extração para processar essas mensagens e realizar a junção das mesmas com as transações SMTP, possibilitando a identificação de usuários denunciados. Dessa forma, foram identificados 67 e 93 *spammers* nas bases de 2009 e 2010, respectivamente. Para todos esses usuários, os endereços IPs e as datas/horas listados nas denúncias coincidiram com dados de transações realizadas, listadas nas bases de tráfego utilizadas.

5. Avaliação e Resultados

Por ser um método iterativo, a avaliação do SpaDeS no que diz respeito à sua sensibilidade à evolução temporal é de extrema importância. Nesta seção, primeiro avalia-se o potencial da nova estratégia de seleção de treino baseada na amostragem ativa (seção 5.1) e, em seguida, avalia-se o benefício de aplicá-la em conjunto com a estratégia original do SpaDeS para a classificação diária de usuários ao longo de 28 dias (seção 5.2).

É importante notar que, em [Las-Casas et al. 2011], o método SpaDeS foi avaliado utilizando 4 classes distintas, sendo duas representantes de usuários legítimos e duas de *spammers*. Ao final, os resultados eram agregados em duas super-classes. Neste trabalho, opta-se por usar diretamente 2 classes como alvo da classificação, uma representando usuários legítimos e outra *spammers*. Uma razão para essa escolha é que o uso do ALAC requer classificação manual dos usuários selecionados e, durante a inspeção manual, a separação dos usuários legítimos e dos *spammers* em duas classes para cada perfil não é trivial. Além disso, para fins de avaliação da eficácia das estratégias de classificação estudadas, todos os usuários da base de 2010 foram manualmente inspecionados e classificados em “legítimo” ou *spammer*. Além disto, utiliza-se também a lista de usuários denunciados como *spammers* nos *logs* de `abuse` de 2010 como informação privilegiada sobre a classe real desses usuários.

Tabela 1. Impacto do Método de Seleção de Treino na Classificação: Taxa de Acerto na Classificação de Usuários Legítimos e *Spammers* e Tamanho do Treino.

	Legítimos Corretamente Classificados	Spammers Corretamente Classificados	Tamanho do Conjunto de Treino
SpaDeS original	97,61%	87,21%	4.212
SpaDeS + seleção ativa (ALAC)	99,16%	84,22%	129

5.1. Seleção Ativa do Conjunto de Treino

Nesta seção avalia-se o potencial da seleção ativa do conjunto de treino, utilizando o ALAC, para a classificação de usuários. Para tal, realiza-se a classificação dos usuários da base de dados de 2010 utilizando, como treino, usuários selecionados da base de 2009. Duas estratégias de seleção foram avaliadas: (1) seleção pelo método ALAC e (2) seleção utilizando a abordagem iterativa. A abordagem iterativa originalmente proposta é aqui considerada para se ter um ponto de comparação a partir do qual se possa analisar a eficácia da seleção ativa.

No primeiro caso, 129 usuários de um total de 5.352 foram selecionados pelo ALAC. A classificação manual de cada um levou à identificação de 57 *spammers* e 72 usuários legítimos. O segundo caso é equivalente ao experimento reportado em [Las-Casas et al. 2011]. O algoritmo X-Means foi usado para agrupar os usuários da base de 2009 em duas classes (legítimo/*spammer*). Os $M = 60$ usuários mais próximos do centróide da classe de legítimos e os 63 usuários reportados nos *logs* de `abuse` coletados no mesmo período da base foram usados como treino para a classificação dos demais usuários daquela base. A partir do resultado dessa classificação e utilizando $\alpha = 20\%$ (algoritmo 1), seleciona-se os usuários que foram classificados com maior confiança para compor o conjunto de treino para a classificação da base de 2010. Usando essa estratégia foram selecionados 4.212 usuários, sendo 3.230 legítimos e 982 *spammers*.

A tabela 1 mostra os resultados da classificação para cada estratégia em termos de percentual de usuários legítimos corretamente classificados, percentual de *spammers* corretamente classificados e tamanho do conjunto de treino selecionado. Note que a seleção ativa produz resultados muitos próximos dos obtidos com a abordagem iterativa. De fato, a seleção ativa levou a uma taxa de acerto de usuários legítimos ligeiramente superior (99,16% versus 97,16%), enquanto que a estratégia original do SpaDeS mostrou-se levemente superior na taxa de acerto dos *spammers* (87,21 % versus 84,2%), mas as diferenças são pequenas. Melhorias em taxas de acertos já elevadas são difíceis de se obter. Vale ressaltar também que advoga-se fortemente que, para um método de detecção de *spammers* na rede de origem, o ideal é que se obtenha a maior taxa de acerto possível para os usuários legítimos (desde que se mantenha uma boa taxa de acerto dos *spammers*), já que o prejuízo causado por falsos positivos pode ser grande, dado que pode trazer desconforto ou penalidade a um usuário legítimo. A classificação errada de alguns *spammers* como legítimos não tem implicações tão fortes, pois os *spams* enviados por eles ainda poderão ser bloqueadas por um filtro no destino ou em algum ponto intermediário.

Em suma, a diferença de desempenho entre ambas as estratégias é pequena, porém a diferença de tamanho dos conjuntos de treino é muito grande. Enquanto a estratégia original precisou utilizar 4.212 usuários, ou seja, 78,20% de todos os usuários da base de 2009, o SpaDeS com seleção ativa precisou de apenas 2,41% (129 usuários). Esses resul-

tados demonstram que o uso do ALAC para seleção do treino é interessante pois: (1) ele leva a bons resultados de classificação e (2) apesar de introduzir um custo extra associado à classificação manual dos usuários selecionados, esse custo é baixo já que o número de usuários selecionados é pequeno. Concluí-se então que o uso da amostragem ativa conjuntamente com o SpaDeS é viável e pode ser uma estratégia promissora para lidar com o impacto da evolução temporal na eficácia da classificação, como será discutido a seguir.

5.2. Impacto da Evolução Temporal na Eficácia da Classificação

Avaliar o comportamento do SpaDeS, particularmente da abordagem iterativa de seleção do conjunto de treino, ao longo do tempo é de extrema importância para se entender se e como o método pode ser aplicado na prática. Por exemplo, a frequência de retreino do modelo de classificação, o que implica na necessidade de novos bons agrupamentos e possivelmente dados externos (p.ex: logs de abuse), depende de quão estável a eficácia da classificação permanece à medida que o tempo passa.

Para avaliar o impacto da evolução temporal na eficácia do SpaDeS foi realizada uma série de experimentos com a base de dados de 2010. Foca-se inicialmente no método original, ou seja, utilizando a abordagem iterativa de seleção de conjunto de treino. O experimento foi projetado como segue. O conjunto de treino inicial foi selecionado dos usuários provenientes do primeiro dia da base: o algoritmo X-Means foi executado para agrupar esses usuários em dois grupos: os $M = 40$ usuários mais próximos do centróide do grupo correspondente à classe de legítimos, juntamente com 43 usuários indicados como *spammers* nos *logs* de `abuse` recebidos nesse dia foram selecionados para compor o conjunto de treino. Esse conjunto de treino inicial foi usado pelo SpaDeS para classificar os usuários do segundo dia. A partir daí, a abordagem iterativa, descrita no algoritmo 1, foi aplicada para a classificação dos usuário nos dias seguintes. Como na seção anterior, foi usado o valor de $\alpha = 20\%$ para selecionar os usuários com maior confiança na classificação do dia i para compor o conjunto de treino do dia $i + 1$.

A figura 1 apresenta a fração de usuários legítimos e *spammers* classificados corretamente em cada um dos 27 dias. Nota-se que, apesar de alguma variação, os resultados obtidos são bastante razoáveis ao longo de todos os 27 dias, mostrando que a estratégia iterativa tende a se manter eficaz durante pelo menos esse período. Como pode ser visto, em todos os dias, o percentual de *spammers* classificados corretamente foi de pelo menos 82%, enquanto que o número de usuários legítimos classificados corretamente, embora menor, permaneça muito bom, superando a marca de 75% em todos os dias.

Em seguida avalia-se se esses resultados poderiam ser melhorados com o uso da seleção ativa do conjunto de treino juntamente com a abordagem iterativa do SpaDeS[2]. Em outras palavras, manteve-se o mesmo conjunto de treino inicial, selecionado a partir da aplicação do X-Means nos usuários do primeiro dia da base. Para os dias seguintes, o conjunto de treino do dia $i + 1$ foi selecionado da seguinte forma. Primeiramente, o algoritmo 1 (com $\alpha = 20\%$) foi aplicado nos resultados da classificação do dia i para selecionar os usuários classificados com maior confiança. O ALAC foi então executado nos dados coletados no dia i para selecionar um conjunto de usuários para classificação

[2]Foram realizados experimentos nos quais a seleção do conjunto de treino incluía também usuários provenientes de fontes externas (`abuse`), em conjunto com as outras estratégias, mas os resultados não foram significativamente diferentes, sendo pois omitidos.

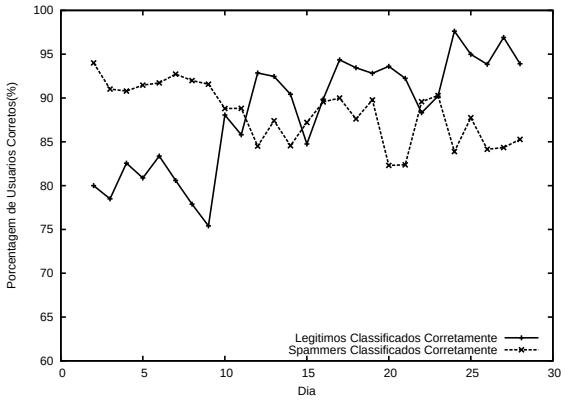
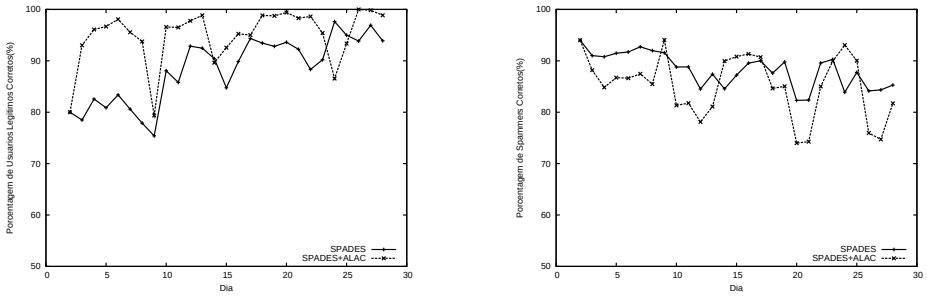


Figura 1. Análise do Impacto Temporal na Eficácia do SpaDeS



(a) Taxa de Acerto de Legítimos (b) Taxa de Acerto de Spammers

Figura 2. Análise do Impacto Temporal: SpaDeS Original versus SpaDeS + Seleção Ativa do Conjunto de Treino.

manual. Dos usuários selecionados pelo ALAC, apenas aqueles que não haviam sido selecionados pela abordagem original do SpaDeS foram inseridos no treino. Busca-se assim reduzir o custo associado à classificação manual exigida pelo ALAC. De fato, observou-se que esse custo é pequeno em todos os 27 dias, com em média apenas 15 e no máximo 26 usuários que devem ser classificados manualmente por dia.

Os resultados são apresentados na figura 2[3]. O uso do ALAC em conjunto com a abordagem iterativa do SpaDeS leva a resultados superiores para classificação de usuários legítimos se comparados com o uso isolado da abordagem iterativa. De fato, a nova estratégia produz resultados iguais ou superiores aos do método original em 93% dos dias analisados. Entretanto, a taxa de acerto na classificação de *spammers* é superior para o SpaDeS original. De fato, a execução de um teste pareado com os 27 resultados diários indicou que as diferenças observadas são significativas com 90% de confiança. Porém, ressalta-se que o uso conjunto do ALAC com o SpaDeS traz um ganho diário médio, em termos de taxa de acerto de legítimos, de cerca de 8%, podendo chegar a 21%, enquanto que a perda na classificação de *spammers* é pequena em todos os dias (3%, em média).

A superioridade na taxa de acerto de legítimos obtida com o uso conjunto do SpaDeS com o ALAC se deve ao fato de que o ALAC, tendo como critério de seleção usuários

[3]As curvas rotuladas como “SpaDeS” correspondem aos resultados apresentados na figura 1, que são aqui repetidos para facilitar a comparação.

com perfis diversos, acaba por incluir no conjunto de treino exemplos de usuários com diferentes perfis legítimos, em termos de uso do protocolo SMTP. A abordagem iterativa, ao contrário, ao selecionar usuários que foram classificados com alta confiança, acaba por incluir no treino usuários com perfis muito semelhantes. Dado que esta é uma classe que exibe grande variabilidade e heterogeneidade [Las-Casas et al. 2011], essa seleção acaba por não capturar padrões legítimos relevantes. Consequentemente, usuários que exibem esses padrões no conjunto de teste acabam sendo confundidos como *spammers*. Por outro lado, a inclusão de vários exemplos de usuários legítimos com diferentes perfis acaba por criar um ruído para a classificação de *spammers*, o que leva a pequena perda de eficácia, quando comparado ao método original.

Finalizando, nota-se que, como já mencionado, objetiva-se minimizar o número de falsos positivos mantendo uma boa taxa de acertos para os *spammers*. Portanto, a melhoria obtida pela utilização da seleção ativa em conjunto com o SpaDeS pode ser considerada significativa, justificando o pequeno esforço manual inserido no processo. Porém, cabe ressaltar que a escolha da estratégia a ser utilizada depende do cenário em questão. O mais comum seria priorizar a classificação correta de usuários legítimos, utilizando então o SpaDeS em conjunto do ALAC. Por outro lado, caso se deseje aumentar ao máximo a taxa de acerto de *spammers*, o método SpaDeS original é a melhor opção.

6. Conclusões e Trabalhos Futuros

Este trabalho avaliou a sensibilidade do método SpaDeS de detecção de *spammers* na rede de origem à evolução temporal. Foi apresentada também uma nova estratégia para seleção de conjunto de treino, baseada em aprendizado ativo, de forma a mitigar os possíveis efeitos temporais na eficácia da classificação. A partir de experimentos com uma base de dados coletada em 2010, mostrou-se que o SpaDeS, apesar de sofrer certa variação, mantém eficácia relativamente estável durante 27 dias. Mostrou-se também que a utilização do SpaDeS em conjunto com a seleção ativa de treino é superior ou se mantém similar à estratégia original, no que tange à classificação correta de usuários legítimos em 93% dos dias, com um ganho diário que pode chegar a 21%, enquanto mantém uma taxa de acerto de *spammers* somente ligeiramente inferior (3%, em média).

Como trabalho futuro, pretende-se validar esses resultados para bases de dados mais novas e possivelmente mais longas.

Referências

Benevenuto, F., Rodrigues, T., Veloso, A., Almeida, J., Gonçalves, M. A., e Almeida, V. (2012). Practical Detection of Spammers and Content Promoters in Online Video Sharing Systems. *IEEE Transactions on Systems, Man and Cybernetics - Part B (to appear)*.

Castilho, L., Las-Casas, P., Dutra, M., Ricci, S., Marques-Neto, H., Ziviani, A., Almeida, J., e Almeida, V. (2010). Caracterização de tráfego SMTP na Rede de Origem. Em *SBRC 2010*, Gramado, Brasil.

Cisco (2010). Cisco Service Control Application for Broadband Reference Guide. Online.

Duan, Z., Gopalan, K., e Yuan, X. (2011). An empirical study of behavioral characteristics of spammers: Findings and implications. *Computer Communications*, 34(14):1764–1776.

Fletcher, D. (2009). A brief history of spam. *Time Magazine*.

Gomes, L., Almeida, V., Almeida, J., Castro, F., e Bettencourt, L. (2009). Quantifying Social And Opportunistic Behavior In Email Networks. *Advances in Complex Systems*, 12(1):99–112.

Gomes, L. H., Cazita, C., Almeida, J. M., Almeida, V., e Jr., W. M. (2007). Workload Models of Spam and Legitimate E-mails. *Performance Evaluation*, 64(7-8):690–714.

Guerra, P. H. C., Pires, D. E. V., Guedes, D., Jr., W. M., Hoepers, C., Steding-Jessen, K., e Chaves, M. (2009). Caracterização de Encadeamento de Conexões para Envio de Spams. Em *SBRC 2009*, Recife, Brasil.

Hao, S., Syed, N. A., Feamster, N., Gray, A., e Krasser, S. (2009). Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine. Em *Proc. Usenix Security*.

Kim, J. e Choi, H. (2008). Spam Traffic Characterization. Em *Int'l Technical Conference on Circuits/Systems, Computers and Communications*, Shimonoseki City, Japão.

Kolcz, A. e Alspector, J. (2001). SVM-based Filtering of E-mail Spam with Content-specifcs Misclassification Costs. Em *Proc. Workshop on Text Mining*, San Jose, EUA.

Lakshmi, R. D. e Radha, N. (2010). Spam Classification using Supervised Learning Techniques. Em *Proceedings of the 1st A2CWiC*, Tamilnadu, India.

Las-Casas, P. H. B., Guedes, D., Almeida, J. M., Ziviani, A., e Marques-Neto, H. T. (2011). Detecção de Spammers na Rede de Origem. Em *SBRC 2011*, Campo Grande, Brasil.

Newman, M. E. J., Forrest, S., e Balthrop, J. (2002). Email Networks and the Spread of Computer Viruses. *Physical Review E*, 66(3):035101.

Ouyang, T., Ray, S., Rabinovich, M., e Allman, M. (2011). Can network characteristics detect spam effectively in a stand-alone enterprise? Em *Proc. 12th Passive and Active Measurement Conference*.

Pelleg, D. e Moore (2000). X-means: Extending K-means with efficient estimation of the number of clusters. Em *17th International Conference on Machine Learning*.

Ramachandran, A. e Feamster, N. (2006). Understanding the Network-Level Behavior of Spammers. *SIGCOMM Computer Communication Review*, 36(4):291–302.

Richard Clayton (2006). spamHINTS: Happily It's Not The Same. Online. http://www.spamhints.org/.

Schatzmann, D., Burkhart, M., e Spyropoulos, T. (2009). Inferring Spammers in the Network Core. Em *Proc. 10th Int'l Conf. on Passive and Active Network Measurement*.

Schryen, G. (2007). *Anti-Spam Measures: Analysis and Design*. Springer.

Shafranovich, Y., Levine, J., e Kucherawy, M. (2010). An Extensible Format for Email Feedback Reports.

Silva, R., Gonçalves, M. A., e Veloso, A. (2011). Rule-based active sampling for learning to rank. Em *Proc. ECML PKDD*, Berlin, Heidelberg.

Sperotto, A., Vliek, G., Sadre, R., e Pras, A. (2009). Detecting Spam at the Network Level. Em *15th Open European Summer School and IFIP TC6.6 Workshop on The Internet of the Future*, Barcelona, Spain.

Taveira, D. e Duarte, O. (2008). Mecanismo Anti-Spam Baseado em Autenticação e Reputação. Em *SBRC 2008*, Rio de Janeiro, Brasil.

Veloso, A., Meira, W., e Zakib, M. J. (2006). Lazy associative classification. Em *Proc. 6th International Conference on Data Mining*, Hong Kong, China.

Venkataraman, S., Sen, S., Spatscheck, O., Haffner, P., e Song, D. (2007). Exploiting network structure for proactive spam mitigation. Em *Proc. 16th USENIX Security Symposium*.

Xie, Y., Yu, F., Achan, K., Gillum, E., Goldszmidt, M., e Wobber, T. (2007). How dynamic are ip addresses? *SIGCOMM Computer Communication Review*, 37:301–312.