UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS QUIXADÁ

TECNÓLOGO EM REDES DE COMPUTADORES

MARIA MICAELE VIEIRA CHAVES

UMA ANÁLISE COMPARATIVA ENTRE TÉCNICAS DE DETECÇÃO DE SPAM COM BASE NA CAPTURA DO TRÁFEGO DA REDE

QUIXADÁ

2017

MARIA MICAELE VIEIRA CHAVES

UMA ANÁLISE COMPARATIVA ENTRE TÉCNICAS DE DETECÇÃO DE SPAM COM BASE NA CAPTURA DO TRÁFEGO DA REDE

Monografia apresentada no curso de Redes de Computadores da Universidade Federal do Ceará, como requisito parcial à obtenção do título de tecnólogo em Redes de Computadores. Área de concentração: Computação.

Orientador: Prof. Dr. Arthur de Castro Callado

QUIXADÁ

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C439a Chaves, Maria Micaele Vieira.
Uma análise comparativa entre técnicas de detecção de spam com base na captura do tráfego da rede / Maria Micaele Vieira Chaves. – 2017.
42 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Redes de Computadores, Quixadá, 2017.
Orientação: Prof. Dr. Arthur de Castro Callado.

1. Detecção-mensagens eletrônicas não solicitadas. 2. Correio eletrônico. 3. Analisador de pacotes. I. Título.
CDD 004.6

MARIA MICAELE VIEIRA CHAVES

UMA ANÁLISE COMPARATIVA ENTRE TÉCNICAS DE DETECÇÃO DE SPAM COM BASE NA CAPTURA DO TRÁFEGO DA REDE

Monografia apresentada no curso de Redes de Computadores da Universidade Federal do Ceará, como requisito parcial à obtenção do título de tecnólogo em Redes de Computadores. Área de concentração: Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Arthur de Castro Callado (Orientador)
Campus Quixadá
Universidade Federal do Ceará – UFC

Prof. Me. Marcos Dantas Ortiz
Campus Quixadá
Universidade Federal do Ceará - UFC

Prof. Me. Michel Sales Bonfim
Campus Quixadá
Universidade Federal do Ceará - UFC

À Deus.

Aos meus pais, Freire e Francisca.

À minha irmã e irmão, Girlene e Evandro.

À minha sobrinha, Islla.

Aos meus amigos.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por ter me iluminado neste trabalho e me dado forças quando mais precisei, pois sei que, absolutamente, tudo que conquistei foi por graça Dele. Sem Ele este trabalho não teria sido realizado.

À minha família, especialmente meus pais, minha irmã e meu irmão, por todo o esforço, amor, carinho, compreensão e dedicação que tiveram comigo ao longo desses anos de graduação e de toda a minha vida.

Ao meu encantador namorado Jonas Costa, por todo o seu apoio, carinho, atenção e por não me deixar desistir quando tudo parecia difícil, apenas por confiar em mim. Amo você!

Ao Professor Arthur Callado, que me orientou com toda paciência neste trabalho e por todo o conhecimento que me transmitiu como professor.

Aos professores participantes da banca examinadora Marcos Dantas e Michel Sales, pelo tempo dedicado ao meu trabalho, pelas valiosas colaborações e sugestões. E por todas as lições que me ensinaram durante a graduação, que me fizeram crescer em diversos aspectos.

Aos meus amigos Isac Cavalcante e Tiago Nascimento, que ao longo da graduação foram uma grande fonte de força para mim. Pelas palavras de amor e carinho que sempre me expressaram, pelo ombro para chorar quando eu mais precisei e por todo o apoio e confiança depositados em mim para a realização deste trabalho. Sem vocês eu ainda estaria no primeiro semestre.

À professora Carla Ilane, que além de me orientar na bolsa do PET Tecnologia da Informação, me aconselhou e motivou, contribuindo para meu crescimento acadêmico e pessoal.

Às minhas amigas, amigos e colegas de bolsa, Ana Paula, Daiane, Dálete, Ingryd, Juliana, Kaiane, Luzia, Ralita, Roseli, Stherfanny, Alexsandro, Anderson Lemos, Edney, Júlio, Lucas, Raul, Rômulo, Anderson, Camila, Ciano, Erika, Igor, Italos, Jordy, Mateus Lima, Natanael, Priscila, Rayanne, Otávio, Samuel que colaboraram direta ou indiretamente para minha graduação.

A todos que direta ou indiretamente fizeram parte da minha formação.

"Todas as coisas cooperam para o bem daqueles que amam a Deus"

(Romanos 8, 28)

RESUMO

Com a popularização da Internet e do uso de e-mail, o envio de *spams* se tornou uma prática comum. Diversas técnicas de detecção de *spams* em e-mails existem e vêm sendo estudadas como contramedida aos *spams* e aos malefícios causados por eles. Administradores de redes analisam constantemente o tráfego da rede através da análise de *traces*. Este trabalho tem o objetivo de realizar uma análise comparativa das técnicas de detecção de *spams*, com base na captura do tráfego da rede. Neste trabalho, criamos diferentes *traces* utilizando o *dataset* ENRON, e aplicamos as técnicas de listas negras, palavras-chave e análise de conteúdo utilizando Naive Bayes e SVM na leitura dos *traces*. Concluímos que é possível realizar a detecção de *spams* a partir da captura do tráfego da rede e que a técnica de análise de conteúdo utilizando SVM apresentou a melhor precisão.

Palavras-chave: Detecção de Spam. E-mail. Traces.

ABSTRACT

With the popularization of the Internet and the use of e-mail, the sending of spam has become a common pratice. Several techniques of e-mail spam detection exist and have been studied as countermeasures to spam and to the harm caused by them. Network administrators constantly analyze network traffic through trace analysis. This work has the objective of performing a comparative analysis of spam detection techniques, based on the capture of the network traffic. In this work we create traces using the ENRON dataset and apply blacklists, keywords and content analysis using Naive Bayes and SVM techniques in the traces evaluation. We conclude that it is possible to perform spams detection from network traffic capture and that content analysis technique using SVM presents the best accuracy.

Keywords: Spam Detection. E-mail. Traces.

LISTA DE FIGURAS

Figura 1 – Funcionamento de listas brancas e listas negras 17
Figura 2 – Funcionamento de classificadores de aprendizagem de máquina 18
Figura 3 – Exemplo de SVM com duas classes 20
Figura 4 – Etapas dos Procedimentos Metodológicos 24
Figura 5 – Separação do conjunto de teste e treino 27
Figura 6 – Fluxograma de Funcionamento da criação do *trace* 28
Figura 7 – Quantidade de acertos 35
Figura 8 – Quantidade de falsos positivos 35
Figura 9 – Quantidade de falsos negativos 36
Figura 10 – Porcentagem de precisão 37
Figura 11 – Ferramenta Spamdas 38

LISTA DE TABELAS

Tabela 1 – Resultados da Técnica de Listas Negras 30
Tabela 2 – Média e desvio padrão listas negras 30
Tabela 3 – Resultado da técnica de Palavras-chave 31
Tabela 4 – Média e desvio padrão palavras-chave 32
Tabela 5 – Resultado Classificador Naive Bayes 33
Tabela 6 – Média e desvio padrão Naive Bayes 33
Tabela 7 – Resultado do Classificador SVM 34
Tabela 8 – Média e desvio padrão SVM 34

LISTA DE ABREVIATURAS E SIGLAS

VP	Verdadeiros Positivos
VN	Verdadeiros Negativos
FP	Falsos Positivos
FN	Falsos Negativos

SUMÁRIO

1 INTRODUÇÃO 13
2 FUNDAMENTAÇÃO TEÓRICA 15
2.1 *Spam* 15
2.2 Técnicas de Detecção de *Spams* 16
2.2.1 Listas Brancas e Listas Negras 16
2.2.2 Palavras-Chave 17
2.2.3 Filtragem baseada em conteúdo 17
2.2.3.1 Filtragem baseada em conteúdo utilizando o classificador Naive Bayes . . . 19
2.2.3.2 Filtragem baseada em conteúdo utilizando o classificador SVM 19
2.3 Análise de Tráfego 20
3 TRABALHOS RELACIONADOS 21
3.1 Métodos avançados para controle de *spam* 21
3.2 Análise do tráfego de *spam* coletado ao redor do mundo 21
3.3 Testes de ferramentas open source no combate ao spam 21
3.4 Filtragem de e-mails de spam utilizando diferentes classificadores com técnicas de redução e seleção de características 22
4 PROCEDIMENTOS METODOLÓGICOS 23
4.1 Preparação e leitura dos traces 23
4.2 Implementação das técnicas de detecção de *spams* 25
5 DESENVOLVIMENTO E RESULTADOS 26
5.1 Preparação e leitura dos *trace*s 26
5.2 Implementação da técnica de listas negras 29
5.3 Implementação da técnica de detecção por palavras-chave 31
5.4 Implementação da técnica de análise de conteúdo utilizando o classificador Naive Bayes 32
5.5 Implementação da técnica de análise de conteúdo utilizando o calssificador SVM 32
5.6 Resultados gerais das técnicas 34
5.7 Spamdas V0.1 37
6 DISCUSSÃO 39
7 CONSIDERAÇÕES FINAIS 41

REFERÊNCIAS 42

1 INTRODUÇÃO

Desde o seu surgimento, a Internet tem causado um grande impacto na vida das pessoas e tem se tornado um veículo de comunicação muito importante, desenvolvendo-se para revolucionar a maneira de fazer negócios e disponibilizar informações. Com a Internet a realidade da globalização se torna viável nas diversas áreas da economia e do conhecimento. Por outro lado, com esse novo canal de comunicação diversas práticas indesejadas como *spams* foram absorvidas (ANTISPAM.BR, 2016).

Um dos serviços mais utilizados da Internet tem sido o e-mail. Por ser um dos serviços mais antigos e utilizados, o e-mail é alvo de uma das práticas maléficas da Internet, o *spam*, que ficou famoso ao ser considerado um grande problema para usuários de e-mail (OLIVO; SANTIN; OLIVEIRA, 2015). O *spam* se assemelha a cartas de correntes para obtenção de dinheiro fácil encontradas nas caixas de correio, panfletos distribuídos nas ruas e as ligações telefônicas de empresas oferecendo seus produtos. Uma diferença extremamente relevante, é o fato de que para enviar cartas ou panfletos, o remetente tem que fazer um investimento mais alto por pessoa alcançada, que muitas vezes dificulta o envio de propagandas em grande escala (CERT.BR, 2012).

Com a popularização da Internet e do uso do e-mail, o remetente de cartas ou panfletos conseguiu, de certa forma, a facilidade de enviar propagandas para um número muito maior de pessoas, com a vantagem de investir pouco. Isso se torna um dos motivadores do envio de *spam* (ANTISPAM.BR, 2016). O envio em larga escala de propagandas e outros conteúdos indesejados pode não custar muito para quem envia, entretanto, tem grandes danos tanto para administradores como para outros usuários da rede. A perda de mensagens importantes, o gasto desnecessário de tempo, impacto na taxa de utilização dos enlaces de rede, a má utilização dos servidores e o investimento extra em recursos são algumas das formas como as pessoas e organizações podem ser afetadas pelo envio de *spam* (CERT.BR, 2012).

Com todos os malefícios que os *spams* geram, surge uma busca por contramedidas, como o uso de sistemas anti-*spams*. Diversas técnicas e sistemas anti-*spam* vêm sendo estudados afim de minimizar a disseminação de *spams*, contudo não há uma técnica que possa erradicar a disseminação, visto que os *spammers* (remetentes de *spam*) estão sempre em busca de descobrir novas maneiras para passar por filtros anti-*spams*, o que motiva a pesquisa contínua sobre melhores formas de detecção de *spams* (TAVEIRA et al., 2006).

Para os administradores de rede e provedores de internet, torna-se fundamental

escolher qual a melhor forma de prevenção e detecção de *spams*, uma vez que detectados, as medidas cabíveis podem ser tomadas. Como os administradores de rede estão constantemente avaliando o tráfego da rede, realizar a detecção de *spams* através do próprio tráfego se torna uma solução viável e atraente. A detecção baseada em traces é útil para identificar o percentual de *spam* no tr[afego da rede (e não só no servidor de e-mail). A detecção na captura, também é útil para possibilitar contra-medidas imediatas de segurança, como o bloqueio de remetentes insistentes inclusive para outros serviços além do e-mail.

O objetivo deste trabalho é realizar uma análise comparativa das principais técnicas de detecção de *spams* encontradas na literatura a partir da captura do tráfego da rede. Para isso, foram implementadas as seguintes técnicas: listas negras, palavras chaves e análise de conteúdo, as quais trabalharam em cima de traces de pacotes. Com a implementação das três técnicas foi feita a execução de testes e análise dos resultados a fim de descobrir se as técnicas funcionam com traces de pacotes e verificar qual técnica obteve o melhor resultado em relação a detecção.

Este trabalho está organizado da seguinte forma: no Capítulo 2 são descritos os conceitos utilizados neste trabalho; no Capítulo 3 são mostrados alguns trabalhos relacionados a técnicas de detecção de *spams*; no Capítulo 4 são descritos os procedimentos metodológicos deste trabalho; no Capítulo 5 o desenvolvimento e resultados são descritos; no Capítulo 6 é feita uma discussão sobre o resultados; por fim, no Capítulo 7, são apresentados os objetivos alcançados, a conclusão e possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo descreve os diferentes conceitos utilizados neste trabalho.

2.1 *Spam*

Spam é um termo utilizado para se referir aos e-mails que não foram solicitados e geralmente são enviados para um grande número de pessoas (CERT.BR, 2012).

De acordo com a organização brasileira AntiSpam.br (2016), desde o aparecimento do primeiro *spam* em 1994, a prática do envio de e-mails não solicitados tem sido aplicada com diversos objetivos, utilizando diferentes meios de propagação na rede. Os tipos de *spams* encontrados até o momento são:

- Correntes: texto que geralmente pede para o destinatário repassar a mensagem um determinado número de vezes e em alguns casos alegando acontecer algo se não repassar. O texto pode contar uma história antiga, descrever uma simpatia ou simplesmente desejar sorte.
- Boatos e lendas urbanas: os boatos geralmente propagam histórias alarmantes e falsas que possam sensibilizar o destinatário a continuar a propagação. As lendas urbanas são parecidas com os boatos, porém se diferem pela justificativa utilizada para atrair a atenção do destinatário, atribuindo veracidade aos relatos, como por exemplo: “Aconteceu com o primo do amigo do meu pai...”
- Propagandas: *spams* com conteúdo de propaganda são conhecidos como UCE (*Unsolicited Comercial E-mail*). São mensagens de publicidade que podem envolver produtos, serviços, pessoas e *sites*.
- Ameaças, brincadeiras e difamação: são mensagens contendo ameaças, brincadeiras incovenientes e difamação de amigos ou ex-(maridos, esposas, namorados, namoradas, colegas de trabalho e chefes).
- Pornografia: mensagens com pornografia em seu conteúdo.
- Códigos maliciosos: mensagens enviadas contendo programas que executam ações maliciosas em um computador. Essas mensagens contêm textos que visam utilizar engenharia social para convencer o destinatário a executar o código malicioso anexo. Muitas vezes esses códigos são usados por fraudadores para conseguir dados pessoais.
- Spit e spim: spit refere-se ao “*spam* via *Internet Telephony*”, em que mensagens não

solicitadas se propagam por outros meios, atingindo os usuários dos "telefones IP". Spim refere-se ao "*spams* via *Instant Message*", mensagens que se propagam por meio dos aplicativos de troca de mensagens instantâneas.

- *Spam* via redes de relacionamentos: mensagens não solicitadas que se propagam por meio de redes sociais através das listas de contatos.

Sharma e Yadav (2015) chamam os e-mails legítimos de *ham* e, neste trabalho, os e-mails legítimos terão a mesma nomenclatura.

2.2 Técnicas de Detecção de *Spams*

A principal contramedida contra *spams* é a adoção de sistemas anti-*spams*, que são compostos por técnicas e procedimentos que atuam na prevenção e coibição de *spams* (TAVEIRA et al., 2006). As subsessões a seguir apresentam as principais técnicas que os sistemas anti-*spams* utilizam.

2.2.1 Listas Brancas e Listas Negras

De acordo com Olivo, Santin e Oliveira (2015) o uso de listas de bons (listas brancas) e maus (listas negras) remetentes é o exemplo mais comum de regra para bloqueio de *spam*. Essa técnica consiste em aceitar ou rejeitar todo e-mail, domínio ou endereço IP contido nessas listas, ou seja, tudo que estiver na lista branca será aceito e tudo que estiver na lista negra será rejeitado.

Um exemplo de como as listas funcionam é mostrado na Figura 1, na qual o e-mail que não é *spam* é enviado por um remetente que consta na lista branca do *Mail Transfer Agent* (MTA)[1] de destino, e é entregue diretamente ao *Mail User Agent* (MUA)[2] do usuário. A entrega direta sem qualquer tipo de processamento apresenta uma vantagem no que diz respeito ao custo computacional, porém pode apresentar riscos de segurança, visto que a fonte de origem que é confiável pode estar comprometida por algum vírus, por exemplo. Na Figura 1, a mensagem que é recebida por um dos endereços que constam na lista negra não é repassada para o MUA (OLIVO; SANTIN; OLIVEIRA, 2015).

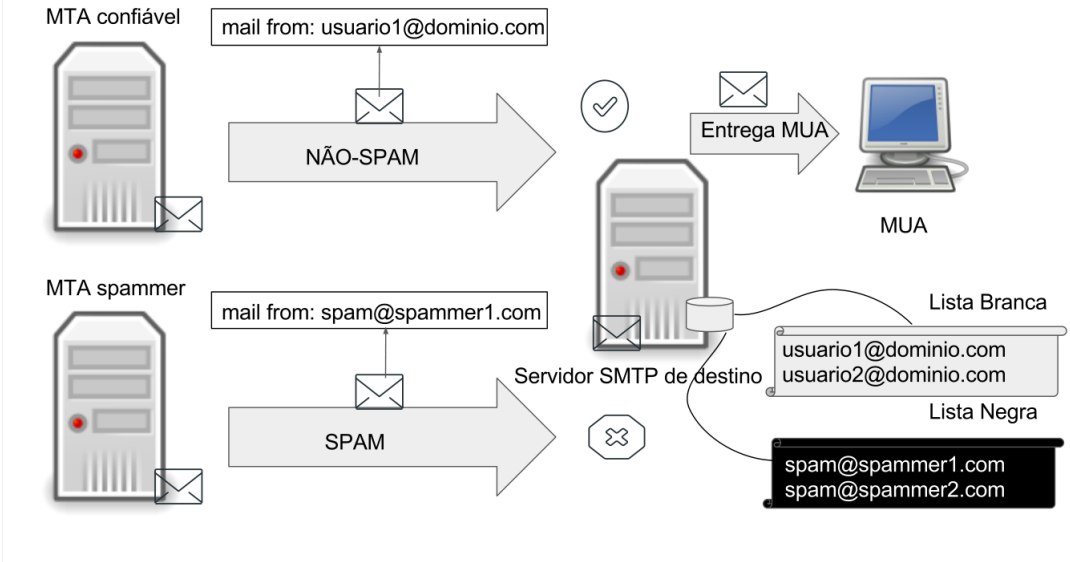
A utilização dessa técnica é bastante precisa e seletiva e as listas precisam se manter sempre atualizadas, porém elas perdem a eficácia quando os *spammers* mudam frequentemente de endereço de e-mail (FABRE, 2005).

---

[1] Servidor responsável pelo envio e recebimento dos e-mails.

[2] Software cliente de serviço de e-mail

Figura 1 – Funcionamento de listas brancas e listas negras



Fonte: Olivo, Santin e Oliveira (2015) - Adaptada pela autora

Neste trabalho, a utilização de listas negras serve apenas para detecção de uma mensagem de e-mail como *spam*, não realizando bloqueio da mensagem caso seja *spam*, visto que o foco do estudo é apenas comparar as técnicas no que diz respeito à detecção.

2.2.2 Palavras-Chave

A técnica de palavras-chave consiste em bloquear e-mails que contenham determinadas palavras no corpo da mensagem. Essas palavras são armazenadas em uma lista, e podem utilizar expressões regulares para identificar algumas variações de *strings* de caracteres, como a palavra viagra, por exemplo, que pode ter a seguinte representação: "V[i1][A4@]GR[A4@]". Entre colchetes estão os caracteres que podem ocorrer em determinada posição da string. Essa técnica deve ser utilizada com cautela, visto que ela não considera o contexto da mensagem e não calcula nenhum tipo de probabilidade. Sendo assim, a inserção de palavras-chave na lista de palavras pode acabar bloqueando mensagens legítimas facilmente (OLIVO; SANTIN; OLIVEIRA, 2015).

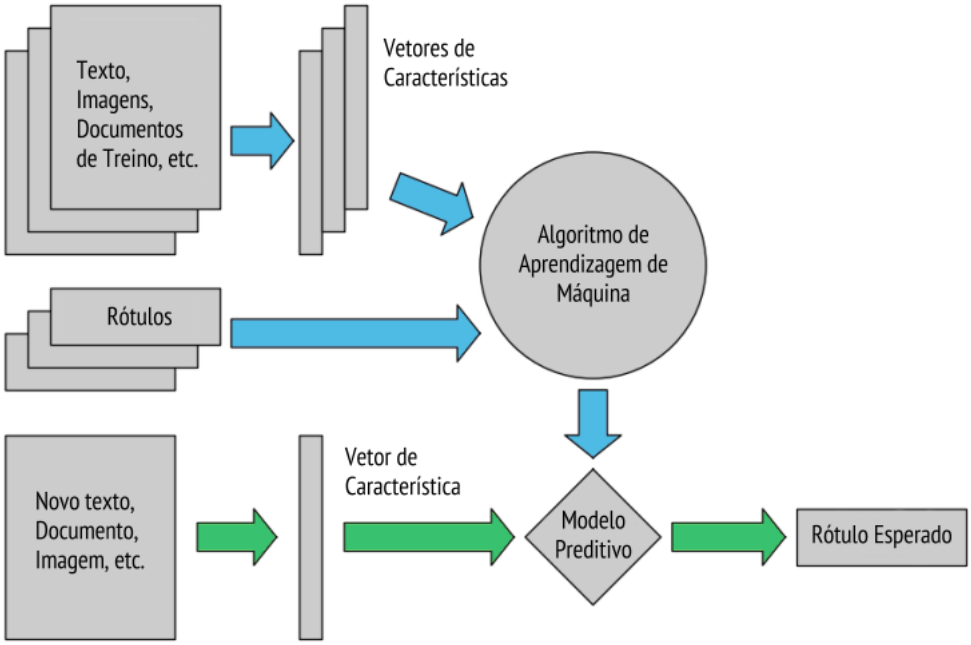
2.2.3 Filtragem baseada em conteúdo

Diferentemente das outras técnicas, em que um conjunto de dados é pré-determinado como possíveis motivos para sinalização de *spam*, a técnica de filtragem baseada em conteúdo pode utilizar a descoberta automática de padrões de dados para a classificação de uma mensagem

como *spam* (OLIVO; SANTIN; OLIVEIRA, 2015). Filtragens baseadas em conteúdo consistem em analisar todo o conteúdo da mensagem, isto é, o texto completo em busca de padrões suspeitos e, com base na identificação de determinados padrões, utiliza estatística e probabilidade para fazer uma classificação do que é ou não *spam* (FABRE, 2005).

Para a realização da filtragem baseada em conteúdo são comumente utilizados algoritmos de aprendizagem de máquina. A classificação da aprendizagem de máquina pode ser feita por um conjunto de algoritmos computacionais que são usados para relacionar objetos a classes específicas. Esses algoritmos são treinados com exemplos que servem como uma linha base. A classificação pode ser aplicada tendo classes específicas pré-definidas, adaptando-se às informações que recebe (DUARTE, 2013). Neste trabalho as classes são: *spam* e *ham*, e são utilizados dois classificadores de aprendizagem de máquina que são explicados nas subseções seguintes. A Figura 2 apresenta o funcionamento dos classificadores utilizados neste trabalho.

Figura 2 – Funcionamento de classificadores de aprendizagem de máquina



Fonte: http://radimrehurek.com/data_science_python/ - Adaptada pela autora

A Figura 2 representa o funcionamento dos classificadores de aprendizagem de máquina deste trabalho, onde inicialmente são necessários arquivos de treinamento com seus respectivos rótulos (dados já classificados). Desses arquivos as características são extraídas pelo algoritmo, que está preparado para receber novos arquivos e classificá-los de acordo com o que foi aprendido com arquivos anteriores. Quando o classificador recebe novos arquivos, ele

apresenta o modelo preditivo, ou seja, a sua classificação, que é comparada depois com o rótulo esperado.

2.2.3.1 Filtragem baseada em conteúdo utilizando o classificador Naive Bayes

O classificador Naive Bayes é popular em filtros de *spams* comerciais e de código aberto. Isso é devido à sua simplicidade, o que o torna fácil de implementar. A complexidade computacional linear do classificador e sua precisão na filtragem de *spams* são comparáveis à de algoritmos de aprendizagem mais elaborados (METSIS; ANDROUTSOPOULOS; PALIOURAS, 2006).

Os Filtros Bayesianos são mais flexíveis do que a filtragem realizada por listas negras, visto que não dependem da identificação do remetente e da manutenção de listas. Se o *spammer* for nômade e disfarçar sua origem, o texto da mensagem pode denunciá-lo. O teorema de Bayes é a base para os filtros bayesianos, sendo um modo de calcular a probabilidade de que um evento irá ocorrer baseado no número de vezes em que o evento ocorreu anteriormente (FABRE, 2005).

O teorema pode ser representado da seguinte forma:

$$P(c|f_1 \dots f_n) = \frac{P(c)P(f_1 \dots f_n|c)}{P(f_1 \dots f_n)}$$

Em que c é a classe e os fns são as características levadas em consideração para a realização da classificação. Para que um elemento seja classificado, é necessário que a probabilidade $P(c|f_1 \dots f_n)$ seja calculada para cada classe c, assim o elemento pertencerá à classe com a maior probabilidade calculada (MEDHAT; HASSAN; KORASHY, 2014).

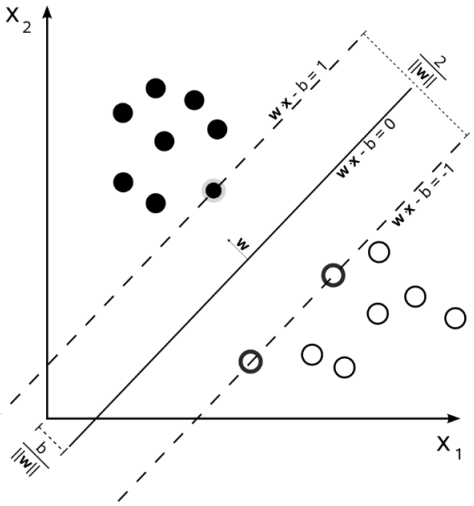
De acordo com Fabre (2005) os sistemas que utilizam esse classificador são treinados, inicialmente, aceitando a entrada de humanos que dizem a qual classe a entrada pertence. Com o tempo, o sistema acumula um banco de informações e, com a aplicação do teorema de *Bayes*, é possível estimar a probabilidade de cada novo dado pertencer a uma classe já classificada. Na detecção de *spams*, o classificador deve detectar *spam* e *ham* , ou seja, o treinamento é feito através de uma amostra de mensagens consideradas *spams* e mensagens consideradas *hams*.

2.2.3.2 Filtragem baseada em conteúdo utilizando o classificador SVM

A Máquina de Vetores de Suporte (*Support Vector Machine* - SVM) é um classificador de aprendizagem de máquina muito utilizado. As SVMs convertem as características relevantes

para a classificação em um ponto disposto em um hiperplano, no qual esse hiperplano é dividido em várias classes, com base em um conjunto de treino inicial. A SVM sempre separa as classes por um hiperplano com a maior distância entre as classes. Dependendo da posição do ponto no hiperplano, perto ou longe de uma classe, é que o classificador irá informar a qual classe pertence a instância de entrada. Existe também uma margem de separação entre as classes que é chamada de vetor de suporte, e os pontos contidos nesta margem são tratados como neutros (DUARTE, 2013). A Figura 3 apresenta um exemplo de SVM com duas classes. O classificador SVM necessita de um conjunto inicial de treino para realizar a aprendizagem, no qual ele vai definir o melhor hiperplano e dividir as classes, para posteriormente classificar as novas entradas de acordo com a classe da qual entrada mais se aproximar.

Figura 3 – Exemplo de SVM com duas classes



Fonte: Duarte (2013)

2.3 Análise de Tráfego

Administradores de rede coletam e monitoram continuamente o tráfego da rede. Com a coleta do tráfego da rede é possível entender as características da rede, ajudando assim no desenvolvimento de novas técnicas que facilitem a otimização dos recursos de rede, colaborando para identificação de padrões de tráfego (FARAH; TRAJKOVIĆ, 2013). Essa análise é realizada com base em *traces* de pacotes. *Traces* de pacotes são arquivos de registros estáticos de diferentes pacotes que trafegam no fluxo da rede (NOTTINGHAM; IRWIN, 2013). Tipicamente, *traces* são capturados em uma interface de rede sob um determinado período de tempo. Neste trabalho, geramos *traces* contendo diferentes e-mails e dentre eles existem *spams* e *hams*.

3 TRABALHOS RELACIONADOS

3.1 Métodos avançados para controle de *spam*

Fabre (2005) estudou técnicas variadas no combate ao *spam*, em especial os filtros *Bayesianos* anti-*spam*, além de apresentar os problemas ocasionados pelos *spams*. As principais técnicas de detecção e ferramentas que implementam as técnicas são apresentadas, bem como a prevenção contra *spams*. Entretanto, os filtros Bayesianos foram o foco do trabalho. De modo semelhante, este trabalho faz um levantamento das principais técnicas de detecção, porém o estudo não irá focar em apenas uma técnica e sim na comparação entre técnicas. Ao invés de apresentar ferramentas existentes que implementem as técnicas a serem comparadas, cada técnica será implementada.

3.2 Análise do tráfego de *spam* coletado ao redor do mundo

Las-Casas et al. (2013) detectaram o tráfego de *spams* de diferentes locais do planeta, utilizando dados que foram coletados por *honeypots* (“potes de mel”, ou iscas para detectar tentativas maliciosas de acesso) e realizaram uma análise que possibilitou a caracterização do tráfego de *spam*, possuindo assim uma visão global e com ela tornando possível mostrar como o tráfego de *spam* se comporta em diferentes locais, evitando possíveis distorções por considerar um único ponto da rede. Nossa proposta também visa detectar o tráfego de *spams*, porém, de modo diferente, analisaremos capturas de tráfego de e-mail para detectar *spams*, ou seja, a detecção de *spams* não será feita em um *honeypot* mas, no tráfego da própria rede. Os dados coletados não serão ao redor do mundo, visto que não pretendemos caracterizar o comportamento do tráfego de *spam*, mas sim detectar o tráfego de *spams* no fluxo da rede.

3.3 Testes de ferramentas open source no combate ao spam

Trentin, Gonzatti e Teixeira (2012) realizaram testes utilizando as ferramentas anti-*spam*: *Bogofilter*, *SpamAssassin* e *SpamPal*. Cada ferramenta foi apresentada, bem como quais técnicas elas utilizam. Os testes visaram verificar a eficácia na classificação das mensagens em cada uma das ferramentas, comparando qual ferramenta na configuração *default* teria o melhor resultado. Propomos também realizar testes para criar uma comparação, porém o que será comparado não serão ferramentas já existentes, mas algumas técnicas de detecção de *spams*

que serão implementadas.

3.4 Filtragem de e-mails de spam utilizando diferentes classificadores com técnicas de redução e seleção de características

Sharma e Yadav (2015) propuseram uma metodologia para detectar um e-mail como *spam* ou *ham* com base na categorização de texto. Eles utilizaram a técnica de PCA (Análise de Componentes Principais) e CFS (Seleção de Característica de Correlação) para redução de características. Várias técnicas para o pré-tratamento do formato de e-mail foram aplicadas, e por fim utilizaram diferentes classificadores para categorizar o e-mail como *spam* ou *ham*. O estudo foca em uma abordagem de filtragem de *spam* baseada em conteúdo usando técnicas de mineração de dados, aplicando as técnicas de PCA e CFS. O estudo pretende aumentar a precisão dos resultados dos classificadores diminuindo o processamento computacional. De modo semelhante utilizamos 2 classificadores para detecção de *spams*, porém não focamos na otimização dos classificadores, tendo em vista que a nossa proposta foca na comparação entre técnicas no que diz respeito à detecção. Utilizamos ainda, o mesmo *dataset* (conjunto de dados) que Sharma e Yadav (2015) usaram, porém utilizamos o *dataset* um pouco maior, enquanto eles utilizaram apenas partes.

4 PROCEDIMENTOS METODOLÓGICOS

Este trabalho foi desenvolvido em 3 macro etapas que estão representadas na Figura 4. Na etapa 1, temos a preparação dos traces. Na etapa 2, é feita a leitura dos traces. Na etapa 3, são implementadas as técnicas de detecção de spam, sendo essa etapa dividida em 4 sub-etapas: implementação da técnica de listas negras (sub-etapa 1); implementação da técnica de palavras-chave (sub-etapa 2); implementação da técnica de análise de conteúdo utilizando Naive Bayes (sub-etapa 3); implementação da técnica de análise de conteúdo utilizando SVM (sub-etapa 4).

Neste trabalho utilizamos as seguintes métricas para realizar a comparação:

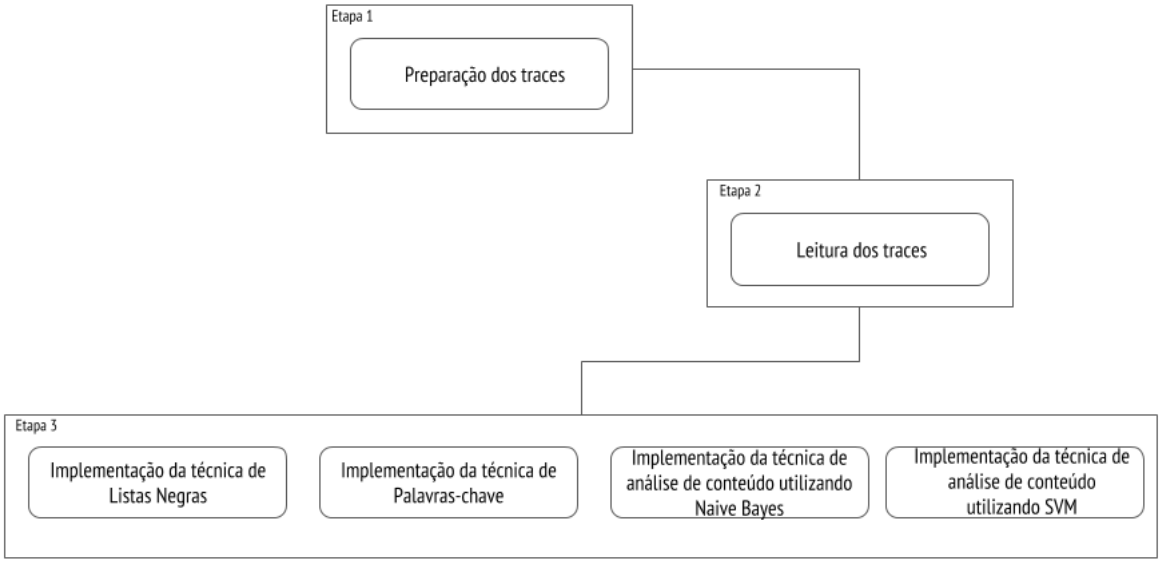
- *Verdadeiros Positivos (VP):* quantidade de e-mails que são *spams* e foram classificados como *spams*.
- *Verdadeiros Negativos (VN):* quantidade de e-mails que são *hams* e foram classificados como *hams*.
- *Falsos Positivos (FP):* quantidade de e-mails que são *hams*, mas foram classificados como *spams*
- *Falso Negativos (FN):* quantidade de e-mails que são *spams*, mas foram classificados como *hams*.
- *Acertos:* soma das quantidades de Verdadeiros Positivos e Verdadeiros Negativos.
- *Precisão:* calculada como a razão entre o número de acertos e o total de e-mails multiplicado por 100: (acertos/total)*100.

Os experimentos foram realizados em 4 computadores executando o sistema operacional Ubuntu 14.04 LTS e com as seguintes configurações de hardware: processador Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz, 8GB de memória *RAM* e 8GB de espaço em disco reservado para *swap*.

4.1 Preparação e leitura dos traces

Para o desenvolvimento deste trabalho, a primeira etapa necessária foi a criação de *traces*, visto que não foi encontrado nenhum *trace* público com um número considerável de e-mails dentre outros tráfegos capturados e que disponibilizasse informações sobre o *trace*. Normalmente, *traces* capturados em redes de acesso ou *backbones* não são públicos ou são muito curtos, contendo uma quantidade ínfima de e-mails. Com o objetivo de gerar os *traces*,

Figura 4 – Etapas dos Procedimentos Metodológicos



Fonte: Elaborada pela autora

encontramos o *dataset* ENRON[1]. Esse *dataset* contém apenas e-mails em formato texto, não sendo um *trace* resultante de uma captura de rede. O *dataset* está dividido em 6 conjuntos: ENRON1, ENRON2, ENRON3, ENRON4, ENRON5 e ENRON6, que são diretórios que contêm arquivos de texto com o conteúdo dos e-mails em inglês. Realizamos o download dos conjuntos no formato de arquivo de texto no seguinte link: <http://www2.aueb.gr/users/ion/data/enron-spam/>.

Inicialmente, foram realizados experimentos com todos os conjuntos. Entretanto, as máquinas utilizadas não conseguiram tratar esse volume de dados. Apesar de possuírem 8GB de memória *RAM*, durante os testes com o *dataset* completo as máquinas finalizavam o processo por estouro de memória. Diante disso, os experimentos foram realizados com metade do conjunto de dados total do *dataset* ENRON, ou seja, 3 conjuntos. Utilizamos os conjuntos 1,2 e 3 e os 3 conjuntos totalizam 16541 e-mails e foram unidos em uma só pasta. Dentre os e-mails desses 3 conjuntos, 4496 são *spams* e 12045 são *hams*, em que o próprio *dataset* informa qual é *spam* e qual é *ham*. Pode-se realizar o download do *dataset* unido em um só diretório no seguinte link: <http://bit.do/dataset1>.

Cada mensagem está em um arquivo de texto separado. O modelo do nome dos arquivos é o seguinte: “0392.2000-02-29.beck.ham.txt”. Os quatro primeiros dígitos indicam a

[1] Conjunto de dados da empresa Enron que foi disponibilizado publicamente para pesquisadores

ordem de chegada dos e-mails, o que não foi levado em consideração neste trabalho, visto que os e-mails são embaralhados antes do envio. Os 8 números seguintes, representam a data de envio no formato aaaa-mm-dd (ano - mês - dia), seguido do nome do dono da caixa de entrada, mais um rótulo informando se é *spam* ou *ham*. Metsis, Androutsopoulos e Paliouras (2006) descrevem detalhadamente esse *dataset*.

A segunda etapa deste trabalho consistiu na leitura dos *traces*, para que posteriormente as técnicas de detecção pudessem trabalhar em cima dos *traces* criados. A leitura dos *traces* tem o objetivo de repassar os e-mails que foram capturados para as técnicas de detecção de *spam*. É importante ressaltar que a captura contém outros pacotes que não são de e-mail, por isso, na leitura dos traces, também é feito esse filtro, de forma a ignorar tráfego que não é SMTP.

4.2 Implementação das técnicas de detecção de *spams*

A terceira etapa do desenvolvimento deste trabalho, foi a implementação das técnicas de detecção de *spams*. Todas as técnicas implementadas neste trabalho operam em cima dos *traces* criados, que foram apresentados em detalhes na Seção 5.1. A técnica de listras negras foi escolhida por ser uma das mais famosas e antigas, encontrada na pesquisa realizada para o desenvolvimento deste trabalho. Na técnica de palavras-chave, as palavras utilizadas foram escolhidas a partir de uma lista de palavras que são comumente encontradas em mensagens de *spam*[2]. Para a técnica de análise de conteúdo, tanto utilizando o classificador Naive Bayes, como utilizando o classificador SVM, nos baseamos no código disponível em (REHUREK, 2017) e o adaptamos às necessidades deste trabalho. Por exemplo, alteramos a leitura dos arquivos de entrada, adicionamos uma função para contagem de e-mails perdidos e fizemos o cálculo de VP, VN, FP, FN. Todas as implementações foram realizadas na linguagem Python.

[2] Palavras retiradas de: <http://bit.do/palavras-chave>

5 DESENVOLVIMENTO E RESULTADOS

Nas etapas 1 e 2 deste trabalho, foram feitas a criação e leitura dos *traces*, respectivamente. Essas etapas são explicadas detalhadamente na Seção 5.1. Nas Seções seguintes, são descritas as formas em que as técnicas de detecção de *spams* foram implementadas.

As implementações deste trabalho foram todas feitas utilizando a linguagem de programação Python.

5.1 Preparação e leitura dos *trace*s

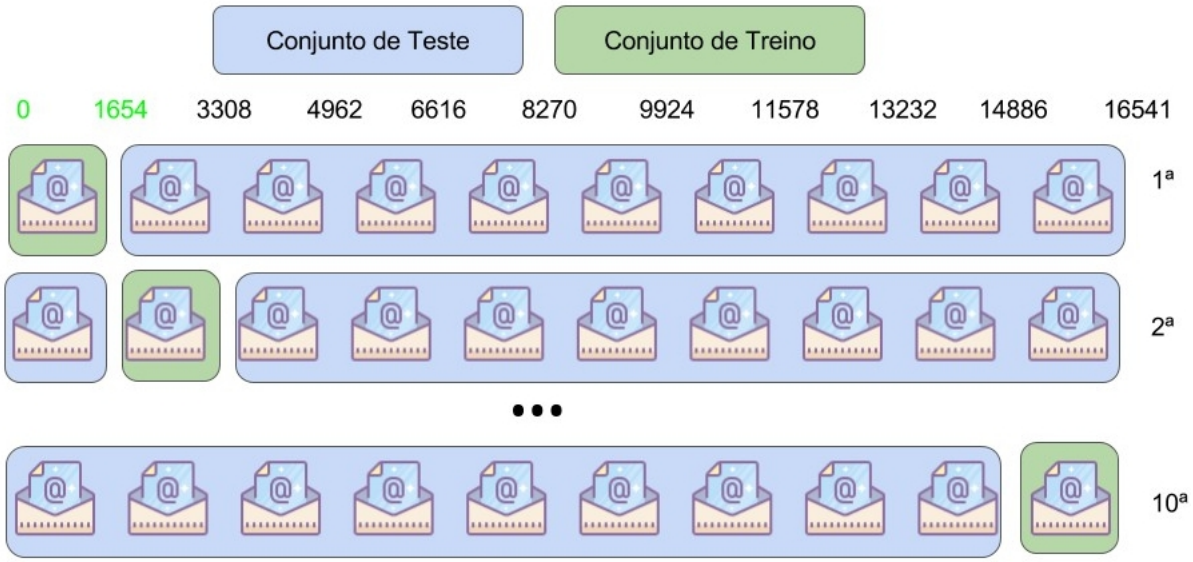
A partir do *dataset* ENRON, realizamos algumas etapas para gerarmos os *traces* utilizados no decorrer do trabalho. Essas etapas são descritas a seguir e podem ser visualizadas na Figura 6:

- ***Separar dataset:*** a princípio, foi utilizada a base de dados em arquivos com formato de texto, visto que se tornou mais viável enviar os e-mails nesse formato. Para manipular o conjunto de dados, foi implementado um módulo "separar_dataset". Esse módulo, inicialmente, embaralha os e-mails do *dataset* e realiza a divisão utilizando a técnica de validação cruzada (*n-fold validation*). Esse *dataset* contém 16541 e-mails. Dividimos esse conjunto de dados em 10 partes, em que cada uma das partes ficou com uma quantidade de 1654 emails, exceto uma, que ficou com a quantidade de 1655. Após divididos os e-mails em 10 partes, foi necessária a separação dos conjuntos de teste (envio e classificação) e treino (aprendizagem). Essa separação foi feita como ilustrado na Figura 5.

  Ainda na Figura 5, podemos visualizar a dinâmica de interação entre os conjuntos de teste e de treino. Inicialmente, é utilizada 1 parte com 1654 e-mails para treino, etapa que chamamos de "aprender" (explicada em detalhes na Seção 5.5), enquanto as outras 9 partes são utilizadas para enviar, totalizando 14887 e-mails . Após o primeiro envio, a parte que antes era de treino será incluída no envio e outra parte é selecionada para treino. Isso é realizado 10 vezes. Sendo assim, temos: aprender0, aprender1... aprender9 e enviar0, enviar1... enviar9.

  Ainda no módulo "separar_dataset", foram gerados os gabaritos de cada envio, que estão no formato "csv". Os gabaritos contêm todos os rótulos dos arquivos que estão sendo enviados. Esses rótulos servem para identificar se o e-mail é *spam* ou *ham*. Incluso no

Figura 5 – Separação do conjunto de teste e treino



Fonte: Elaborada pela autora

gabarito está também o nome dos arquivos. Esses gabaritos servirão posteriormente para a verificação de acertos de cada técnica, visto que será feita a comparação do gabarito com o que cada técnica classificou.

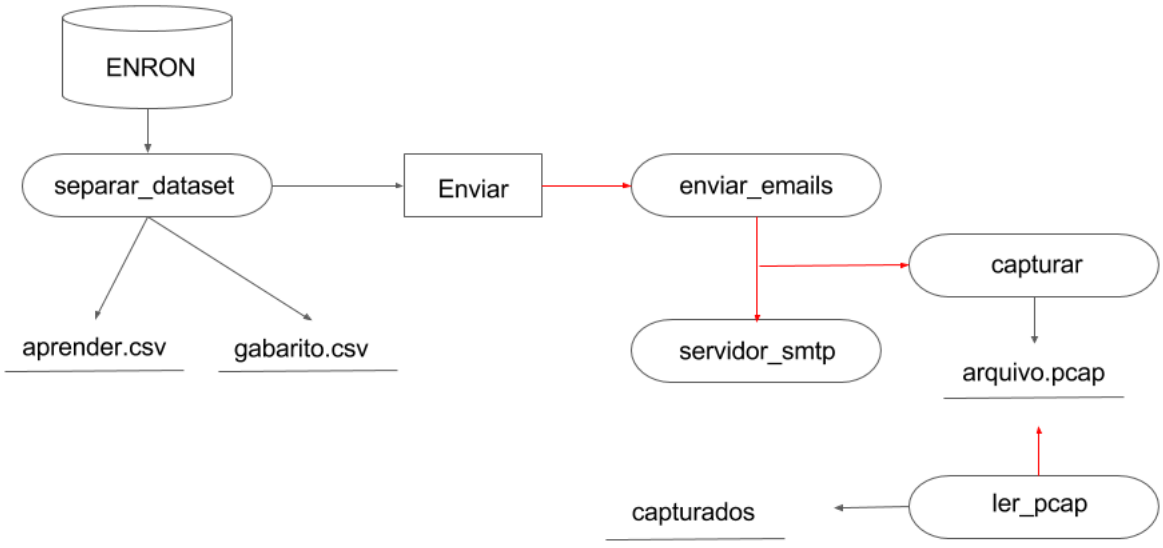
- ***Enviar e-mails:*** implementamos um módulo chamado "enviar_emails" para encaminhar os e-mails separados para envio. Nesse módulo, existe apenas um remetente fictício que envia e-mails de *ham* e vários remetentes fictícios que enviam e-mails de spam, os chamados *spammers*. Para o recebimento de todas as mensagens, temos um único destinatário fictício. Os remetentes fictícios não alteram o funcionamento e resultado das técnicas, em razão de que as técnicas visam apenas classificar se é *spam* ou não. Esses e-mails são todos recebidos por um servidor de e-mail local, que foi feito através de uma implementação genérica em linguagem *Python* e utiliza a porta 25 para comunicação. Aleatoriamente, alguns endereços de *spammers* enviam *hams* e o endereço fictício que envia e-mails legítimos envia alguns *spams*.
- ***Captura do tráfego:*** Existem diversos programas capazes de realizar a captura do tráfego da rede, como por exemplo, *Wireshark* e *Tcpdump*. Entretanto, consideramos que a utilização de tais programas dificultaria a integração com os outros módulos desenvolvidos neste trabalho. Desse modo, foi implementado um módulo chamado "capturar", que realiza a captura de pacotes no tráfego da rede, sendo basicamente um *sniffer* implementado em linguagem *Python*, utilizando a biblioteca "scapy". Esse módulo armazena em um arquivo 'pcap' todos os pacotes que foram capturados, formando assim os *traces* que este trabalho

manipula. Esses *traces* serão utilizados para todas as técnicas.

Neste trabalho, a captura foi realizada na mesma máquina do servidor de e-mails em dois processos distintos, apenas para fins de demostração, mas, em um cenário real, o módulo de captura poderia ser executado em um *firewall*, por exemplo.

- **Leitura do *trace***: Após a captura e armazenamento dos pacotes, um outro módulo chamado "ler_pcap" realiza a leitura do arquivo 'pcap'. Nessa leitura, é feita a contagem de quantos pacotes SMTP o *trace* possui, a ordenação dos pacotes SMTP pelo número de sequência (a fim de montar as mensagens de e-mails), a remoção de pacotes duplicados e ainda a verificação de se são e-mails que possuem texto, visto que as técnicas que são utilizadas neste trabalho lidam apenas com texto e remetentes. Esse módulo de leitura gera, ainda, o arquivo 'capturados.csv' que será manuseado posteriormente quando as técnicas de detecção forem executadas, tendo seus rótulos comparados com os rótulos do arquivo 'gabarito.csv'. Como são 10 *traces*, temos: capturados0, capturados1...capturados9. O conteúdo desses arquivos são: o conteúdo do e-mail, remetente, destinatário e o nome do arquivo original. Esses traces estão disponíveis para download[1].

Figura 6 – Fluxograma de Funcionamento da criação do *trace*



Fonte: Elaborada pela autora

A Figura 6 ilustra o funcionamento da criação dos *traces*. O módulo "separar_dataset"

[1] <http://bit.do/trace-pcap>

separa o *dataset* ENRON e gera a pasta 'Enviar', o arquivo 'aprender' e 'gabarito'. A pasta 'Enviar' serve de entrada para o módulo "enviar_emails". Esse módulo, por sua vez, envia os arquivos da pasta 'Enviar' para o servidor local, no caso, o "servidor_smtp" que é uma implementação genérica na linguagem *Python*. Enquanto os e-mails estão sendo enviados, o módulo "capturar" está executando para realizar a captura dos pacotes, e, após ele ser encerrado, gera o 'arquivo.pcap'. O módulo "ler_pcap" realiza a leitura do 'arquivo.pcap' e gera o arquivo 'capturados.csv'. Vale ressaltar que os módulos "enviar_emails", "servidor_smtp" e "capturar" são executados na mesma máquina simultaneamente, em três processos distintos. A execução da criação do *trace*, a partir do módulo enviar ocorreu 10 vezes, com diferentes conjuntos de e-mails, para que assim obtivéssemos 10 *traces* distintos.

Para que a implementação de cada técnica tivesse o controle de acertos, foi necessária a comparação do arquivo 'capturados.csv' com o 'gabarito.csv'. Percebemos perdas de alguns e-mails durante a captura. Provavelmente, a grande quantidade de e-mails enviados em um curto período de tempo (1 e-mail a cada 0.1 segundo) tenha ocasionado essas eventuais perdas, devido ao grande tráfego de pacotes gerado. Quando algum pacote SMTP era perdido, não era possível montar o e-mail correspondente e, devido à transferência de e-mails ser local, uma quantidade grande de pacotes pode também ter passado muito rapidamente pelo servidor, de forma a inviabilizar a captura pelo módulo de captura.

Os e-mails que foram perdidos, não poderiam ser comparados com o gabarito original que contém todos e-mails enviados. Portanto, organizamos os e-mails capturados na mesma ordem do gabarito, visto que após a captura eles ficam desorganizados, e, após a ordenação, excluímos do gabarito as entradas que não estão no *trace*. As execuções para a criação dos *traces* foram realizadas em diferentes máquinas, havendo divergência no número de pacotes perdidos. A ordenação dos e-mails capturados na mesma ordem do gabarito foi realizada na implementação de cada técnica.

5.2 Implementação da técnica de listas negras

Para a implementação da técnica de listas negras, o código desenvolvido utiliza um arquivo de texto chamado de "lista_negra", que contém endereços de e-mails de *spammers*. Esses endereços são fictícios, criados como base em uma lista dos *spammers* mais famosos[2].

A partir da leitura dos *traces* (detalhada na Seção 5.1), comparamos se o remetente

[2] https://en.wikipedia.org/wiki/List$_o f_s$*pammers*

do e-mail está na lista de *spammers*. Se ele estiver, a mensagem é classificada como *spam*, caso contrário, é classificado como *ham*. Vale ressaltar que a leitura dos *traces* gera os arquivos "capturados0.csv", "capturados1.csv" ... "capturados9.csv", e esses arquivos são utilizados como entrada para a técnica.

Os resultados obtidos nessa técnica são armazenados em um arquivo de texto chamado "resultados". Como neste trabalho utilizamos 10 *traces*, a técnica foi executada para cada um deles, totalizando 10 execuções. Os resultados obtidos são apresentados na Tabela 1. Essa tabela apresenta o número da execução, o total de e-mails enviados, o total de e-mails perdidos durante a captura dos pacotes, o total de acertos, os verdadeiros negativos, os verdadeiros positivos, os falsos positivos, os falsos negativos e a precisão.

Tabela 1 – Resultados da Técnica de Listas Negras

Execução	Total de E-mails	E-mails Perdidos	Acertos	VN	VP	FP	FN	Precisão
1	14887	3517	9064	6608	2456	1681	625	79.72%
2	14887	2058	10248	7437	2811	1849	732	79.88%
3	14887	1902	10450	7603	2847	1848	687	80.48%
4	14887	3481	9098	6635	2463	1700	608	79.77%
5	14887	2314	10024	7341	2683	1856	693	79.73%
6	14887	2077	10262	7470	2792	1845	703	80.11%
7	14887	1323	10830	7885	2945	2001	733	79.84%
8	14887	3508	9034	6588	2446	1664	681	79.39%
9	14887	3493	9083	6639	2444	1665	646	79.72%
10	14886	3201	9365	6839	2526	1685	635	80.15%

Fonte: Elaborada pela autora

A Tabela 2 apresenta a média e o desvio padrão das execuções para: acertos, FP, FN e precisão.

Tabela 2 – Média e desvio padrão listas negras

	Acertos	FP	FN	Precisão
Média	9745,8	1779,4	674,3	79,879%
Desvio Padrão	686,61	115,53	43,87	0,30%

Fonte: Elaborada pela autora

5.3 Implementação da técnica de detecção por palavras-chave

Para a implementação da técnica de detecção por palavras-chave, primeiramente cadastramos uma lista de palavras comumente encontradas em e-mails de *spam*[3]. Como explicado na Subseção 2.2.2, algumas letras podem apresentar variações pelos *spammers*, como por exemplo a palavra 'offers', que poderia ter algumas dessas combinações: '0ffers', 'Off3rs', '0ff3r5'. Nesse caso, desenvolvemos a técnica para que estivesse preparada para esse tipo de variação de caracteres. Além disso, também definimos que as palavras poderiam sofrer uma alteração com espaços entre letras e pontos entre letras, como por exemplo a palavra "offer" poderia ter essas variações: 'o f f e r', 'o.f.f.e.r', 'o. f. f. e. r'.

A partir da leitura dos *traces* (explicada na Seção 5.1), verificamos em cada e-mail se uma das palavras cadastradas, ou variações das palavras cadastradas, estão presentes nos emails extraídos da leitura. Caso a palavra esteja presente no e-mail, ele é considerado como *spam*, caso contrário, é classificado como *ham*.

Os resultados obtidos nessa técnica são armazenados em um arquivo de texto chamado "resultados" e, como usamos 10 *traces*, a técnica foi executada para cada um, totalizando 10 execuções. Os resultados obtidos são mostrados na Tabela 3. Essa tabela apresenta o número da execução, o total de e-mails enviados, o total de e-mails perdidos durante a captura dos pacotes, o total de acertos, os verdadeiros negativos, os verdadeiros positivos, os falsos positivos, os falsos negativos e a precisão.

Tabela 3 – Resultado da técnica de Palavras-chave

Execução	Total de E-mails	E-mails perdidos	Acertos	VN	VP	FP	F.N	Precisão
1	14887	3517	7915	6264	1651	2025	1430	69.61%
2	14887	2058	8928	7024	1904	2262	1639	69.59%
3	14887	1902	9055	7127	1928	2324	1606	69.73%
4	14887	3481	7949	6299	1650	2036	1421	69.68%
5	14887	2314	8776	6957	1819	2240	1557	69.8%
6	14887	2077	8945	7057	1888	2258	1607	69.83%
7	14887	1323	9451	7484	1967	2402	1711	69.68%
8	14887	3508	7904	6226	1678	2026	1449	69.46%
9	14887	3493	7905	6243	1662	2061	1428	69.38%
10	14886	3201	8151	6461	1690	2063	1471	69.76%

Fonte: Elaborada pela autora

A Tabela 4 apresenta a média e o desvio padrão das execuções para: acertos, FP, FN

[3] Palavras retiradas de: <http://bit.do/palavras-chave>

e precisão.

Tabela 4 – Média e desvio padrão palavras-chave

	Acertos	FP	FN	Precisão
Média	8497,9	2169,7	1531,9	69,652%
Desvio Padrão	591,29	142,08	105,07	0,14%

Fonte: Elaborada pela autora

5.4 Implementação da técnica de análise de conteúdo utilizando o classificador Naive Bayes

Como explicado no capítulo 2, na Subseção 2.2.3.1, o classificador Naive Bayes necessita de um conjunto de treino. Neste trabalho, como foram criados 10 conjuntos diferentes de envio de e-mails, que são explicados na Seção 5.1, existem, também, 10 conjuntos de treino, que provêm do *dataset* ENRON e estão nos arquivos "aprender0.csv", "aprender1.csv" ... "aprender9.csv". Através desses arquivos, em cada execução da técnica, que ela é executada para cada *trace*, o classificador irá realizar a aprendizagem para posteriormente realizar a classificação dos e-mails que foram capturados e se encontram nos arquivos "capturados0.csv", "capturados1.csv" ... "capturados9.csv". Após a classificação dos e-mails, é feita a comparação com o gabarito, em que cada execução da técnica contém seu gabarito correto e é através dos gabaritos que controlamos os acertos.

Os resultados obtidos nessa técnica são armazenados em um arquivo de texto chamado 'resultados' e são mostrados na Tabela 5. Essa tabela apresenta o número da execução, o total de e-mails enviados, o total de e-mails perdidos durante a captura dos pacotes, o total de acertos, verdadeiros negativos, verdadeiros positivos, falsos positivos, falsos negativos e precisão.

A Tabela 6 apresenta a média e o desvio padrão das execuções para: acertos, FP, FN e precisão.

5.5 Implementação da técnica de análise de conteúdo utilizando o calssificador SVM

Como explicado no Capítulo 2, na Subseção 2.2.3.2, o classificador SVM necessita de um conjunto de treino inicial e este trabalho opera em cima de 10 conjuntos diferentes de e-mails,

Tabela 5 – Resultado Classificador Naive Bayes

Execução	Total de E-mails	E-mails perdidos	Acertos	VN	VP	FP	FN	Precisão
1	14887	3517	9609	8287	1322	2	1759	84.51%
2	14887	2058	10482	9283	1199	3	2344	81.71%
3	14887	1902	10839	9447	1392	4	2142	83.47%
4	14887	3481	9813	8331	1482	4	1589	86.03%
5	14887	2314	10708	9190	1518	7	1858	85.17%
6	14887	2077	10657	9313	1344	2	2151	83.19%
7	14887	1323	11374	9884	1490	2	2188	83.85%
8	14887	3508	9165	8251	914	1	2213	80.54%
9	14887	3493	9718	8295	1423	9	1667	85.29%
10	14886	3201	9902	8522	1380	2	1781	84.74 %

Fonte: Elaborada pela autora

Tabela 6 – Média e desvio padrão Naive Bayes

	Acertos	FP	FN	Precisão
Média	10226,7	3,6	1969,2	83,85%
Desvio Padrão	684,53	2,54	266,36	1,69%

Fonte: Elaborada pela autora

que são explicados na Seção 5.1. Existem 10 conjuntos de treino que derivam do *dataset* ENRON, e estão armazenados nos arquivos ‘aprender0.csv”, “aprender1.csv” . . . “aprender9.csv”. Por meio desses arquivos, em cada execução da técnica, o classificador irá realizar a aprendizagem, para posteriormente realizar a classificação dos e-mails que foram capturados, ou seja, os e-mails que estão nos *traces*. Após a leitura do *trace*, que é explicada na Seção 5.1, os e-mails são armazenados nos arquivos “capturados0.csv”, “capturados1.csv” . . . “capturados9.csv”. Para cada um dos arquivos “capturados.cvs” o classificador é executado. Após a classificação, é feita a comparação com os gabaritos, que contém os rótulos de cada e-mail que foi enviado. É importante lembrar que a sincronização do gabarito com os e-mails que foram capturados é feita antes que ocorra a comparação visto que, durante a captura, alguns e-mails foram perdidos e não teria como compará-los.

Os resultados obtidos nessa técnica são armazenados em um arquivo de texto chamado ‘resultados’ e são mostrados na Tabela 7. Essa tabela apresenta o número da execução, o total de e-mails enviados, o total de e-mails perdidos durante a captura dos pacotes, o total de acertos, os verdadeiros negativos, os verdadeiros positivos, os falsos positivos, os falsos negativos e a precisão.

A Tabela 8 apresenta a média e o desvio padrão das execuções para: acertos, FP, FN e precisão.

Tabela 7 – Resultado do Classificador SVM

Execução	E-mails perdidos	Total de E-mails	Acertos	VN	VP	FP	FN	Precisão
1	14887	3517	10803	7865	2938	424	143	95.01%
2	14887	2058	12357	9036	3321	250	222	96.32%
3	14887	1902	12564	9181	3383	270	151	96.76%
4	14887	3481	10939	7998	2941	337	130	95.91%
5	14887	2314	12132	8872	3260	325	116	96.49%
6	14887	2077	12168	8866	3302	449	193	94.99%
7	14887	1323	13090	9604	3486	282	192	96.51%
8	14887	3508	10954	8019	2935	233	192	96.27%
9	14887	3493	10922	7993	2929	311	161	95.86%
10	14886	3201	11265	8224	3041	300	120	96.41%

Fonte: Elaborada pela autora

Tabela 8 – Média e desvio padrão SVM

	Acertos	FP	FN	Precisão
Media	11719,4	318,1	162	96,053%
Desvio Padrão	833,12	70,41	36,12	0,61%

Fonte: Elaborada pela autora

5.6 Resultados gerais das técnicas

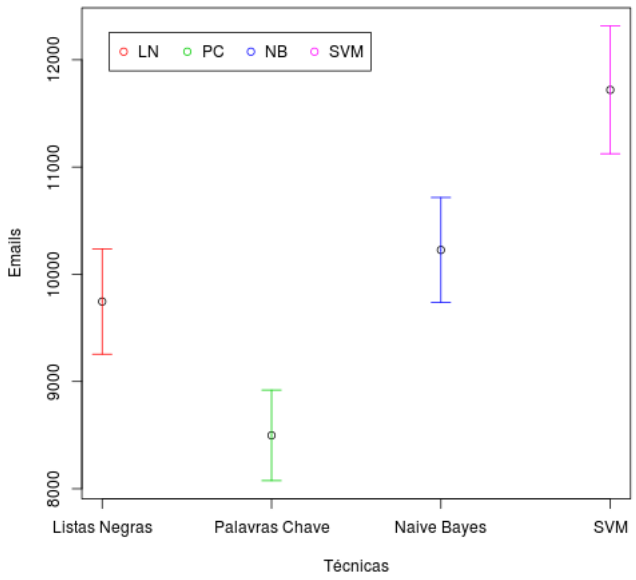
As figuras a seguir ilustram os resultados para: acertos, falsos positivos, falsos negativos e precisão de cada uma das técnicas. O nível de confiança utilizado foi 95%. Para o cálculo do intervalo de confiança, baseado no nível de confiança estabelecido, foi utilizada a ferramenta R[4].

A Figura 7 ilustra a quantidade de acertos para cada uma das técnicas, utilizando intervalos de confiança calculados para um nível de confiança de 95%. Nela, podemos ver que a técnica de análise de conteúdo utilizando SVM apresentou uma maior quantidade de acertos em comparação com as demais técnicas. Logo em seguida está a mesma técnica de análise de conteúdo, porém utilizando o classificador Naive Bayes. A técnica de listas negras vem em 3º lugar. Apresentando um número baixo de acertos em comparação com as demais técnicas, está a técnica de palavras-chave.

A Figura 8 apresenta a quantidade de falsos positivos para cada uma das técnicas baseada em um nível de confiança de 95%. Pode-se observar que a técnica de análise de conteúdo utilizando o classificador Naive Bayes apresentou um número muito baixo de falsos positivos. A técnica de análise de conteúdo utilizando o classificador SVM apresentou um número baixo de

[4] Disponível em: http://www.r-project.org

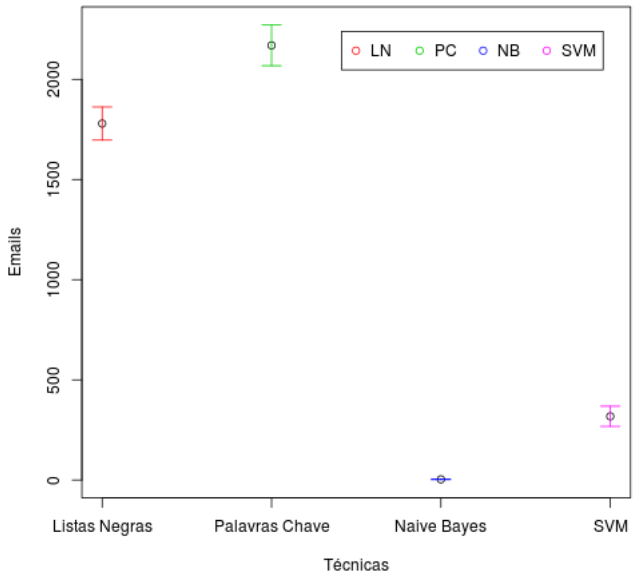
Figura 7 – Quantidade de acertos



Fonte: Elaborada pela autora

falsos positivos, porém bem maior do que o Naive Bayes. A técnica de listas negras apresentou um número elevado de falsos positivos, porém foi menor do que a técnica de palavras-chave, que apresentou o índice mais elevado de falsos positivos.

Figura 8 – Quantidade de falsos positivos

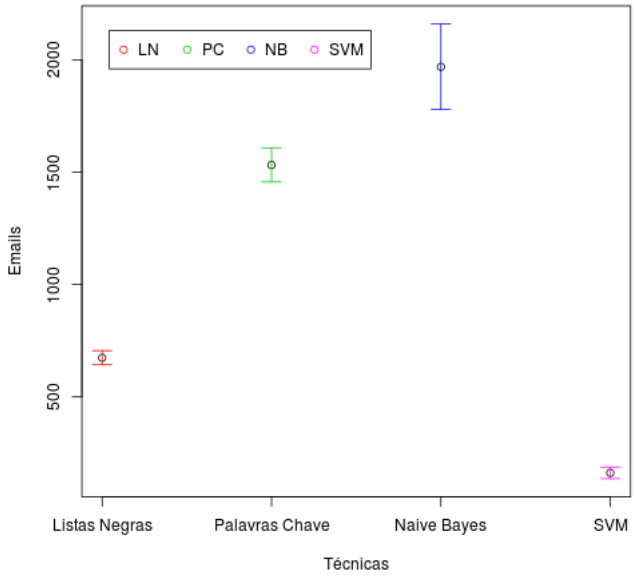


Fonte: Elaborada pela autora

A Figura 9 expõe a quantidade de falsos negativos para cada técnica baseada em um

nível de confiança de 95%. É possível notar que a técnica de análise de conteúdo utilizando Naive Bayes, apresentou um alto índice de falsos negativos. A técnica de palavras-chave também apresentou um alto índice de falsos negativos, porém menor do que o classificador Naive Bayes. A técnica de listas negras apresentou um baixo nível de falsos negativos, porém um pouco maior do que o classificador SVM. A técnica de análise de conteúdo utilizando SVM foi a que apresentou o menor índice de falsos negativos.

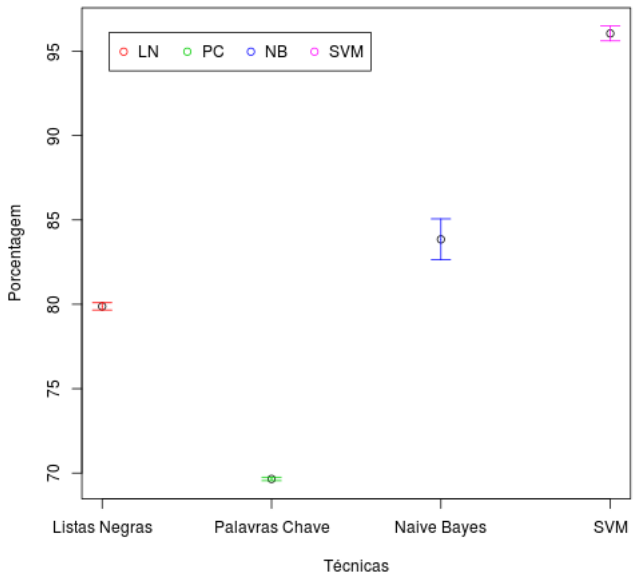
Figura 9 – Quantidade de falsos negativos



Fonte: Elaborada pela autora

A Figura 10 mostra a precisão de cada técnica baseada num nível de confiança de 95%. É possível perceber que a técnica de análise de conteúdo utilizando SVM tem uma precisão alta. O classificador Naive Bayes também apresentou uma precisão alta, mas com um intervalo de confiança maior. A técnica de listas negras apresentou um índice de precisão um pouco menor do que o classificador Naive Bayes. A técnica de palavras-chave apresentou uma precisão muito baixa em comparação com as demais técnicas.

Figura 10 – Porcentagem de precisão



Fonte: Elaborada pela autora

5.7 Spamdas V0.1

Todas as técnicas implementadas neste trabalho, através de diferentes módulos, foram unidas em uma só ferramenta que recebeu o nome de 'Spamdas'. A ferramenta recebeu o nome 'Spamdas' devido à utilização da biblioteca *Pandas*[5]. A ferramenta realiza desde a leitura de um *trace* até a execução de todas as técnicas. As técnicas também podem ser executadas uma por vez. A ferramenta completa está disponível em: <github.com/micaelevieira/spamdas/>. É importante lembrar que ela utiliza o *dataset* ENRON, e ainda está em fase de desenvolvimento, visto que sua primeira versão foi desenvolvida para o escopo deste trabalho, precisando ainda ser adaptada a cenários reais, sem gabarito.

A Figura 11 apresenta a versão 0.1 da ferramenta Spamdas, que foi desenvolvida neste trabalho. Com essa ferramenta, é possível ler um *trace* que tenha a extensão ".pcap" e separar os e-mails para posteriormente utilizar algumas das técnicas de detecção explanadas neste trabalho. Vale ressaltar que esta ferramenta foi inteiramente desenvolvida para este trabalho e ainda precisa ser adaptada para cenários reais.

[5] <http://pandas.pydata.org/>

Figura 11 – Ferramenta Spamdas

Fonte: Elaborada pela autora

6 DISCUSSÃO

Com os experimentos realizados neste trabalho foi possível constatar que, as técnicas de detecção de *spams* utilizadas funcionam com traces de pacotes, visto que foi possível utilizar as técnicas, e estas, por sua vez, apresentaram resultados semelhantes aos encontrados na literatura. Este fato se torna interessante visto que empresas e instituições podem realizar a detecção de *spams* não somente em seu servidor de e-mail, mas em alguma outra máquina configurada para ter acesso ao tráfego da rede.

Como ilustrado na Tabela 2, a média da precisão da técnica de listas negras chegou a 79,879%, o que pode ser considerada uma precisão boa. Entretanto, em um cenário real, essa precisão pode conter uma grande variação, visto que os *spammers* mudam constantemente de endereços, que demandará constante atualização de tais listas. A média da taxa de falsos positivos foi um pouco elevada, o que torna a técnica perigosa, posto que várias mensagens que poderiam ser importantes não seriam lidas por seus usuários finais.

Aa Tabela 4 mostra que a técnica de palavras-chave apresentou uma média de precisão de 69,652%, o que não é uma precisão muito boa. A técnica apresentou ainda médias elevadas de falsos positivos e de falsos negativos, o que torna a técnica tanto perigosa como desagradável, visto que, além dos usuários perderem várias mensagens que poderiam importantes, também teriam que ler várias mensagens de *spam*.

Na Tabela 6, a média da precisão da técnica de análise de conteúdo utilizando Naive Bayes foi de 83,85%, sendo considerada uma boa precisão. A média da taxa de falsos positivos foi bem baixa (3,6), o que a torna uma técnica boa, considerando que os usuários finais perderiam quase nada das mensagens importantes. Algo que poderia tornar-se desagradável para usuários finais é o fato da média dos falsos negativos ter sido relativamente alta (média de 1969,2 por experimento, ou 16,15%) com Naive Bayes, possivelmente porque maioria dos e-mails eram *hams*.

A média de precisão da técnica de análise de conteúdo utilizando SVM, que é apresentada na Tabela 8, foi de 96,053, o que é considerada uma ótima precisão. A média de falsos positivos foi de 70,41, sendo maior que a taxa de falsos negativos que foi de 36,1, o que torna a utilização um pouco preocupante, em vista de que mais mensagens legítimas estão sendo perdidas do que usuários estão vendo *spams*.

No geral, a técnica de listra negras se apresentou melhor do que a técnica de palavras-chave. Entretanto, se apresentou abaixo da técnica de análise de conteúdo. A técnica de análise

de conteúdo, tanto utilizando Naive Bayes quanto utilizando SVM, apresentou uma boa precisão, sendo que a média da precisão utilizando SVM foi melhor (96,053%), enquanto a média da precisão do Naive Bayes foi apenas razoável (83,85%). É importante lembrar que, apesar do classificador SVM ter apresentado uma taxa de precisão melhor que o Naive Bayes, ele apresentou uma taxa de falsos positivos bem maior que o Naive Bayes, ou seja, para um usuário que recebe várias mensagens importantes, o classificador Naive Bayes seria mais adequado, visto que ele perderia menos mensagens importantes. Por outro lado, o classificador Naive Bayes apresentou um nível elevado de falsos negativos em comparação com o SVM, sendo assim, a utilização do Naive Bayes acarretaria que usuários veriam muitos *spams*, enquanto que se usassem o SVM, veriam menos.

7 CONSIDERAÇÕES FINAIS

Neste trabalho, apresentamos as principais técnicas de combate ao *spam* que são utilizadas na literatura, sendo que não foi encontrada qualquer abordagem que realizasse a classificação de *spams* sobre arquivos de captura da rede. Tal abordagem é relevante tanto para medir a incidência de SPAMs no tráfego de um *backbone* que serve várias empresas quanto no tráfego de uma empresa que hospeda serviço de e-mail de terceiros (sem acesso administrativo aos servidores), como um *datacenter*. Para os experimentos, foram criados *traces* de pacotes contendo e-mails de *spam* e *ham* através do envio e captura dos e-mails presentes no *dataset* ENRON. As técnicas implementadas neste trabalho foram: listas negras, palavras-chave e análise de conteúdo, utilizando os classificadores Naive Bayes e SVM. Foi desenvolvida, ainda, uma ferramenta chamada Spamdas, que se encontra em fase desenvolvimento e está disponível no repositório público GitHub[1]. Os *traces* criados neste trabalho se encontram disponíveis para download[2]

Conseguimos alcançar o objetivo da execução das técnicas, que funcionaram sobre os arquivos de captura de tráfego da rede. Realizou-se ainda uma análise comparativa entre os resultados destas técnicas.

Na comparação entre as técnicas, em relação à métrica de acertos, a técnica de análise de conteúdo utilizando SVM apresentou o melhor resultado, visto que apresentou o maior número de acertos. Com relação à precisão, a técnica que apresentou o melhor resultado também foi a de análise de conteúdo utilizando SVM.

Em relação a métrica de falsos positivos, a técnica de análise de conteúdo utilizando Naive Bayes apresentou o melhor resultado, posto que a taxa de falsos positivos foi muito baixa. Na métrica de falsos negativos, a técnica de análise de conteúdo utilizando SVM apresentou mais uma vez o melhor resultado.

Neste trabalho, não foi mensurado o impacto de *spams* no tráfego da rede, sendo assim, sugerimos como trabalho futuro avaliar essa perspectiva, realizando análises sobre os arquivos de captura que contém *spam* junto ao tráfego normal da rede.

---

1 <https://github.com/micaelevieira/spamdas>

2 <http://bit.do/traces-capturas>

REFERÊNCIAS

ANTISPAM.BR. **Motivadores de envio de spam**. 2016. Acesso em: 13 jul. 2017. Disponível em: <http://antispam.br/conceito/>.

CERT.BR, C. de Estudos Resposta e Tratamento de Incidentes de Segurança no B. **Cartilha de Segurança para Internet**. 2ª edição. ed. [S.l.]: Comitê Gestor da Internet no Brasil, 2012. ISBN 978-85-60062-54-6.

DUARTE, E. S. **Sentiment analysis on twitter for the portuguese language**. Tese (Doutorado) — Faculdade de Ciências e Tecnologia, 2013.

FABRE, R. C. **Métodos Avançados para Controle de Spam**. [S.l.]: Campinas, 2005.

FARAH, T.; TRAJKOVIĆ, L. Anonym: A tool for anonymization of the internet traffic. In: IEEE. **Cybernetics (CYBCONF), 2013 IEEE International Conference on**. [S.l.], 2013. p. 261–266.

LAS-CASAS, P. H. B.; GUEDES, D.; JR, W. M.; HOEPERS, C.; STEDING-JESSEN, K.; CHAVES, M. H.; FONSECA, O.; FAZZION, E.; MOREIRA, R. E. Análise do tráfego de spam coletado ao redor do mundo. **Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos – SBRC**, 2013.

MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: A survey. **Ain Shams Engineering Journal**, Elsevier, v. 5, n. 4, p. 1093–1113, 2014.

METSIS, V.; ANDROUTSOPOULOS, I.; PALIOURAS, G. Spam filtering with naive bayes-which naive bayes? In: **CEAS**. [S.l.: s.n.], 2006. v. 17, p. 28–69.

NOTTINGHAM, A.; IRWIN, B. A high-level architecture for efficient packet trace analysis on gpu co-processors. In: IEEE. **2013 Information Security for South Africa**. [S.l.], 2013. p. 1–8.

OLIVO, C. K.; SANTIN, A. O.; OLIVEIRA, L. E. S. Abordagens para detecção de spam de e-mail. **XV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais**, 2015.

REHUREK. **Practical Data Science in Python**. [S.l.], 2017. Rare Technologies. Disponível em: <http://radimrehurek.com/data_science_python/>. Acesso em: 10 mai. 2017.

SHARMA, A. K.; YADAV, R. Spam mails filtering using different classifiers with feature selection and reduction technique. In: IEEE. **Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on**. [S.l.], 2015. p. 1089–1093.

TAVEIRA, D. M.; MORAES, I. M.; RUBINSTEIN, M. G.; DUARTE, O. Técnicas de defesa contra spam. **Livro Texto dos Mini-cursos do VI Simpósio Brasileiro de Segurança da Informaç ao e de Sistemas Computacionais**, p. 202–250, 2006.

TRENTIN, M. A. S.; GONZATTI, A.; TEIXEIRA, A. C. Testes de ferramentas open source no combate ao spam. **Revista CIATEC-UPF**, v. 4, n. 2, p. 24–34, 2012.