UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Detecção Automática de *Phishing* em Páginas Web

Janainny Sena Carvalho

Manaus - Amazonas
Julho de 2013
Janainny Sena Carvalho

Detecção Automática de *Phishing* em Páginas Web

Proposta de mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Informática.
Área de concentração: Redes de Computadores.

Orientador: Prof. Dr. Eduardo James Pereira Souto
Co-Orientadora: Profª. Dra. Eulanda Miranda dos Santos
Janainny Sena Carvalho

Detecção Automática de *Phishing* em Páginas Web

Proposta de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Informática.
Área de concentração: Redes de Computadores.

Banca Examinadora

_____

Prof. Dr.

_____

Profª.

_____

Prof.

Manaus – Amazonas
Maio de 2013

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A Internet vem servindo como infraestrutura para a disponibilização uma ampla variedade de recursos e serviços, incluindo suporte a correio eletrônico, comércio eletrônico, Internet banking, mídias sociais, entre outros. Toda essa popularização tem tornado a rede mundial cada vez mais relevante na vida diária das pessoas e das organizações, crescendo sua importância na atividade social e econômica global.

Nesse contexto, vem crescendo a cada momento o número de usuários conectados a Internet. Em 2007, aproximadamente 28,7% da população mundial tinha acesso a Internet, em torno de 1,96 milhões de usuários. Atualmente esses números são superiores a 77,8 milhões de usuários acessando a rede mundial de computadores [Zhang *et al.,* 2007].

Entretanto, essa massiva utilização de serviços *online*, passou a ser alvo frequente de diversos agentes que exploram recursos computacionais de forma não autorizada. Nesse sentido, serviços oferecidos pela Internet podem trazer consigo as fraudes eletrônicas como *spam*, códigos maliciosos *(malwares*), ataques de negação de serviços (DDoS - *Distributed Denial of Service*), vírus, *worms* e *phishing*.

Infelizmente, a demanda cada vez maior por novos recursos, a fim de prover mais serviços e funcionalidades para os usuários da Internet, principalmente os serviços de comércio eletrônico e Internet banking tem promovido ainda mais o crescimento de ataques *phishing*. Pesquisas recentes indicam que os ataques *phishing* se mantém no topo das listas das maiores vulnerabilidades em aplicações web nos últimos anos, conforme as estatísticas divulgadas em [APWG, 2010] [Zhang, Jianyi et al,. 2011] .

Phishing é uma forma de fraude que combina a engenharia social e técnicas de falsificação de páginas *web* com intuído de roubar informações confidencias como senhas, *logins*, número de cartões de crédito entre outras informações que são consideráveis críticas e sensíveis [Basnet, 2008]. Na Internet, o *phishing* pode atingir ao usuário (vítima) de várias maneiras, através de uma

janela *pop-up* no navegador (*browser*), uma URL maliciosa, de mensagens instantâneas ou através de mails recebidos. Geralmente, a vítima é convencida a executar uma ação (por exemplo, clicar num link), que poderá levar a instalação de algum *malware* ou ao redirecionamento do usuário a um site malicioso.

Em decorrência da grande quantidade de ataques *phishing*, diferentes abordagens e técnicas para solucionar ou mitigar o problema têm sido propostas nos últimos anos. Dentre elas, técnicas baseadas no emprego de listas negras (blacklists) [Ma, Justin et al., 2009] [Ye, Cao et al., 2009], de técnicas de aprendizagem e mineração de dados [Whittaker et al., 2009] [Miayamoto et al., 2009] [Fette et al., 2009] e na análise da estrutura estática e dinâmica dos elementos do aplicação web [Miayamoto et al., 2009] [Basnet ae al., 2008].

Entretanto, apesar das contribuições desses trabalhos, o problema ainda figura no topo das principais listas de vulnerabilidade, fato que motiva a pesquisa em busca de novas técnicas que possam contribuir com soluções de prevenção, detecção ou contenção de ataques *phishing*.

Este trabalho apresenta e avalia um método que emprega técnicas supervisionadas de aprendizagem de máquina, que têm como vantagem, a descoberta de padrões a partir de um conjunto de fatos ou observações rotuladas, induzindo a máquina a um processo de aprendizagem.

O método apresenta foco especializado na detecção de ataques *phishing*, permitindo uma exploração mais detalhada do problema e a análise de um conjunto de características relevantes que são extraídas de exemplos obtidos de bases reais da Internet e submetidos a métodos de aprendizagem estáveis e amplamente usados em problemas de classificação, tais como o *Naive Bayes*, SVM e Árvore de Decisão.

A estratégia proposta é estruturada em fases que tem o objetivo de viabilizar a extração de um conjunto de características que são relevantes para detecção de *phishing* em páginas *web,* essa extração ocorreu com a análise do conteúdo estático da URL, ou seja, características extraídas da URL, do documento que compõe a *web* e das extraídas a partir de informações contidas na base de dados *online*.

1.1 Motivação

A preocupação com a segurança da informação tem sido cada vez mais destacada pelos desenvolvedores de aplicações *web*, principalmente devido aos inúmeros serviços que são disponibilizados por estes na Internet.

Essa preocupação é justificada principalmente pela sofisticação e aumento crescente das técnicas utilizadas para roubar dados na web, em especial dados de transações comerciais.

A fraude *phishing* se tornou a partir de 2009 responsável por 66% dos ataques realizados em páginas *web,* foi identificado neste ano que apenas um grupo de golpista foi responsável por dois terços de todos ataques *phishing* lançados na Internet e que também em 2009, *phishing* foi considerado uma avalanche e cada vez mais grupos fraudulentos vem ser aperfeiçoando para realizar novas tentativas de ataques *phishing*. [APWG], [Aaron Greg, 2010].

Nesse contexto, apesar de existirem várias soluções propostas para detecção de ataques *phishing*, assim como implementações para prevenção desse tipo de ameaça, *Anti-PhishingWork Group* [APWG, 2010] identificou cerca de 20 mil sites novos de *phishing* nos meses de julho a dezembro de 2008.

Ainda vale ressaltar que fatos que comprovam a "eficiência" do *phishing* podem ser observados diariamente. Assim se destaca a empresa de consultoria Gartner Inc. [2009] ao divulgar que, no ano de 2008, os criminosos *phishers* causaram um prejuízo de mais de 1,7 bilhões de dólares nos Estados Unidos. Ainda, a China Daily [2011], jornal *on-line* da empresa de segurança Beijing Rising Information Technology Co denuncia que páginas *phishing* roubaram cerca de 3 bilhões de dólares na China, em 2010. No Brasil, foram contabilizadas 31.008 tentativas de frades *phishing* em 2010 [CERT.Br,2011].

Nesse sentido, a implementação de técnicas para identificar automaticamente sites *phishing,* por exemplo, é algo que devemos dar a devida atenção. Inclusive algumas técnicas já são utilizadas por grandes sites, tais como: Google, eBay, Paypal. Uma das técnicas que esses sites utilizam é a classificação automática observando as características que um site *phishing* apresenta. Assim, um site que apresentar certas características pode ser classificado como um site *phishing* e, a utilização de classificadores de aprendizagem de máquina nessa análise é fundamental, uma vez que é possível ensinar padrões para que esses sejam classificados automaticamente.

Contudo, diante das ameaça que a fraude *phishing* vem impactando, diversas abordagens técnica tem sido proposta para que de algum modo seja realizada a detecção de *phishing* em páginas *web*. Porém, sempre existirá a necessidade de se obter novas ferramentas ou mecanismos para que esses possam ajudar na identificação de *phishing,* uma vez que a cada dia surgem novas formas de realizar fraudes em páginas *web*.

1.2 Objetivos do Trabalho

O objetivo deste trabalho é desenvolver um mecanismo de detecção de *Sites Phishing* em páginas *web* que utilize técnicas de aprendizagem de máquina e mineração de dados. Primeiramente, serão construídas as características e também a base de dados. Posteriormente, a extração das características mais relevantes serão selecionadas e extraídas através de testes em diversos classificadores e avaliadores. Por fim, serão aplicados algoritmos de aprendizagem de máquina, tais como *Naive Bayes, SVM* e Árvore de Decisão, para a classificação e análise das taxas obtidas.

Os objetivos específicos são os seguintes:

1. Desenvolver técnicas para detecção de *phishing* em páginas *web*.
2. Identificar base de dados com sites *phishing* e legítimos de diversos repositórios;
3. Definir principais características de sites *phishing* que sejam relevantes no processo de detecção de sites *phishing* propostas na literatura;
4. Analisar as características selecionadas, utilizando algoritmos de aprendizagem de máquina, tais como: *Naive Bayes*, SVM e Àrvore de Decisão;
5. Avaliar a relevância das características, dentro da amostra geral, com objetivo de melhorar a desempenho dos algoritmos de aprendizagem e reduzir o custo computacional.

1.3 Contribuições

Conforme os objetivos e metodologia utilizada neste trabalho foi possível produzir as seguinte contribuições:

1. Demonstrar um método para detectar *phishing* em páginas *web*, a partir da utilização de técnicas de aprendizagem de máquina e com essas identificar padrões de sites *phishing*.
2. Estabelecer e analisar um conjunto de características que identifiquem sites *phishing* em páginas *web*, e que ainda possam serem aplicadas em diversos contextos que envolvam outras soluções complementares;
3. Expor uma análise comparativas dos resultados obtidos com os métodos de classificação *Naive Bayes*, SVM e Árvore de Decisão, como intuito de apresentar as taxas obtidas nesse

processo de classificação, e a partir desse resultado demonstrar o desempenho obtido de *Phishing* em páginas *web*;

4. Disponibilizar a base de dados utilizada para emprego em futuros trabalhos de pesquisa na área de aprendizagem de máquina.

1.4 Organização do Documento

O desenvolvimento desta dissertação, a partir do Capítulo 2, está organizado como segue:

- O Capítulo 2 apresenta informações fundamentais para a compreensão do tema deste trabalho. Serão detalhados os conceitos básicos de detecção de *phishing*, formas de propagação e detecção, aprendizagem de máquinas e, principalmente, o funcionamento básico do classificador *Naive Bayes* SVM e Árvore de Decisão.
- O Capítulo 3 apresenta os trabalhos relacionados ao tema proposto, abordando sobre: Listas, Filtros, Heurísticas e trabalhos que utilizaram aprendizagem de máquina para detecção de sites *phishing* em páginas *web*.
- O Capítulo 4 descreve as etapas e o método proposto, também será detalhada a grupo de características selecionada para detecção de *phishing* em páginas *web*, a base de dados utilizada ;
- O Capítulo 5 descreve todos experimentos realizados com o conjunto de características e classificadores de aprendizagem de máquina utilizado neste trabalho, e com resultados obtidos, é demonstrado uma análise e comparação com outros desfechos mencionados em outras literaturas. E por fim, um resumo dos resultados e conclusões obtidas tomando por base a literatura consultada e a pesquisa no que diz respeito às direções futuras.

Capítulo 2

Conceitos Básicos e Definições

Neste capítulo, serão definidos os conceitos de *phishing,* suas categorias, tipos de técnicas que estão sendo empregadas no combate dessa fraude, Por fim, as ultimas seções define os conceitos sobre aprendizagem de máquina e descreve os classificadores usados nos experimentos deste trabalho.

2.1 Phishing

De acordo com Richard [2005] o termo *phishing* foi sugerido para descrever o roubo de senha e contas da American On Line (AOL) em 1996. Desde então, na primeira década do século XXI, a definição de *phishing* tem se expandido, incluindo atacantes que usam vetores de ataque tais como:e-mail, cavalo de troia e *keyloggers* para enganar as vítimas [Garera *et al.* 2006].

Outros autores , afirmam que *phishing* pode ser também enquadrado na forma de estelionato que usa engenharia social para fazer vítimas, as quais são enganadas geralmente com o objetivo de obter suas informações pessoais (predominando interesse de cunho financeiro) [Olivo, 2012] [Ma, Justin et al., 2009] [Sheng, Steve, et al., 2009]. Esse aspecto pode ser interpretado de acordo com o Código Penal Brasileiro, considerando que estelionato é "obter, para si ou para outrem, vantagem ilícita, em prejuízo alheio, induzindo ou mantendo alguém em erro, mediante artifício, ardil, ou qualquer outro meio fraudulento" [Código Penal Brasileiro, Título II, Cap. VI, Art. 171].

Normalmente os ataques *phishing* podem ser realizados por meio do envio de uma mensagem de *e-mail* que contém *links* para um *site* ilegítimo com semelhanças ao *site* original. Esses e-mails costumam incluir mensagens com um sentido de urgência, apontando a necessidade da operação para os destinatários fornecerem informações confidenciais, como já mencionados anteriormente ou os usuários podem simplesmente digitar a página no barra de seu navegador e ser direcionado para uma página ser falsa. A figura 2.1 abaixo mostra uma página legitima e outra página falsificada.

A Figura 2.1 mostra um site legítimo e um site *phishing*, para destacar a semelhança entre ambos.

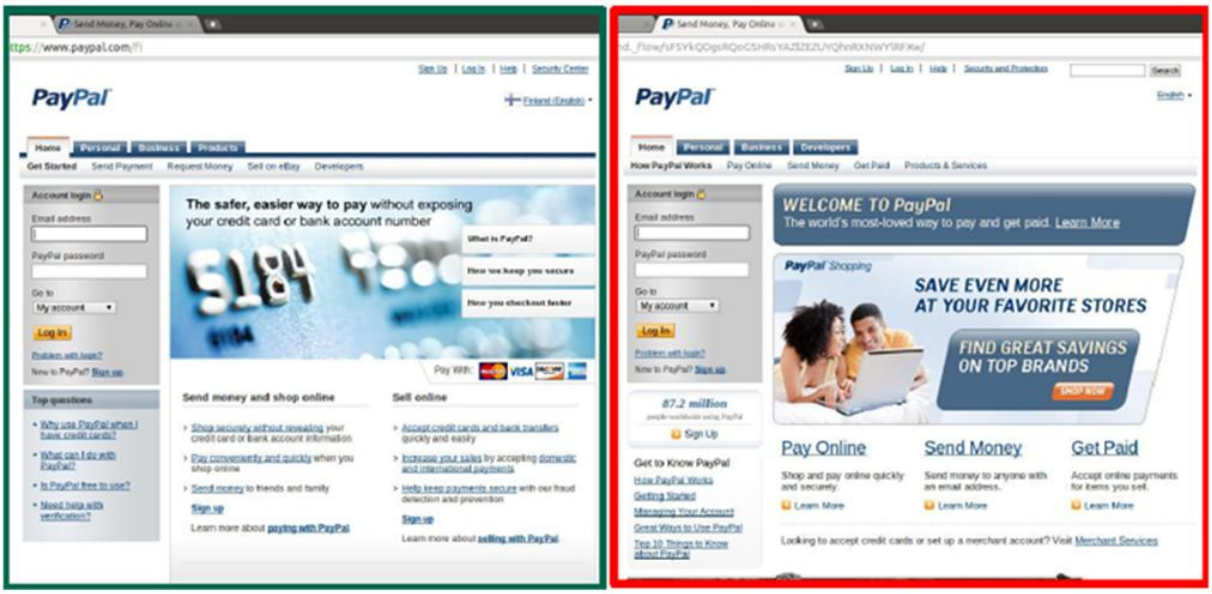

Figura 2.1: (a) página legítima, (b) página *phishing.*

O usuário quando acessa indevidamente um site *pshishing*, normalmente não consegue identificar, pois o construtor de uma página *phishing*, intitulado de *phisher,* elabora o site de forma que este seja o mais idêntico possível ao site original, dificultando a percepção dos usuários em distinguir páginas originais e fraudulentas.

O *phisher* também utiliza a *web*, forjando as páginas com interesse em obter informações. Dessa forma, um site tenta enganar o usuário disfarçando a URL, na tentativa de parecer ao máximo com o real, por exemplo: www.paypal.com. e www.sitefalso.com. O usuário poderia achar que está acessando o endereço do paypal, mas na realidade ele estaria acessando o site falso (www.sitefalso.com). Existem outras formas de burlar a atenção dos usuários, o que dificulta a identificação de sites fraudulentos, pois os golpistas constroem sites cada vez mais parecidos com os originais. Para melhor ilustração, tem-se a seguir um enquadramento situacional:

2.2 Formas de Propagação

Alguns ataques *phishing* conseguiram convencer em torno de 5% dos seus destinatários, de maneira a fornecer informações subjetivas para sites falsos [Loftedness, 2004], o que foi constatado por Litan [2004] ao afirmar que, em 2003, cerca de dois bilhões de usuários forneceram suas informações para sites falsos, resultando em perdas em torno de US$ 1,2 bilhões para os EUA .

Conforme Downs *et al*. [2006], os ataques *phishing* são mais bem sucedidos quando o *phisher* é capaz de manipular os usuários. Portanto, é importante compreender tipos de modelos mentais que as pessoas utilizam para ler um *e-mail* ou páginas *Web.* Neste caso, o desenvolvimento de uma melhor compreensão dos motivos que levam as pessoas a caírem em sites fraudulentos é imprescindível. *Phishers* exploram a diferença entre o modelo do sistema e o modelo mental dos usuários com o intuito de melhor ludibriar a vítima [Wu, 2006].

A maior parte dos usuários desconhece a segurança fornecida pelo navegador *Web* e os canais de obtenção de informações de segurança. Um ataque *phishing* é bem sucedido porque os usuários parecem tomar decisões imprudentes ao navegar num site sem as devidas precauções de segurança.

Um estudo realizado por Dhanija *et al*. [2006] sustentou que muitos usuários têm problemas na detecção de ataque *phishing*, pois não conseguem 'distinguir um site legítimo de um site fraudulento'. Esse estudo identificou que o melhor site *phishing* conseguiu enganar 90% dos usuários (participantes).

Um ataque típico de *phishing* ocorre quando uma determinada quantidade *de e-mail phishing* é enviada aleatoriamente para os usuários. Esses e-mails são camuflados para serem confundidos com e-mails originais de tal forma que a vítima que os recebe pode ser facilmente convencida que o e-mail está vindo de uma organização legítima, como uma instituição bancária, por exemplo [Rosiello, 2007].

Assim, um ataque *phishing* acorre quando o usuário recebe um *e-mail* com *link* de um *site*, porém, um *site phishing*. Esse *e-mail* pode ser, por exemplo, de agências bancárias, Serasa etc., *sites* que normalmente solicitam informações sigilosas, tais como: número do CPF, número do cartão de crédito, *login,* entre outras informações confidencias.

Segundo Fette *et al*. [2007], muitos usuários caem na armadilha da retenção de dados sigilosos, em diversas operações, fornecendo informação vulnerável às ações fraudulentas.

Outro tipo de ataque *phishing web* é definido como *pharming*, processo através do qual o atacante desvia o usuário para direcioná-lo a sites fraudulentos [Abu-Nimeh *et al.* 2007]. Neste tipo de fraude, ao tentar se conectar a um *site Internet Backing* digitando o endereço do *site* em seu *browser*, por exemplo, o usuário será redirecionado para a página falsificada, normalmente idêntica ao *site* legítimo. Em geral, esse tipo de ataque ocorre devido ao sequestro ou envenenamento do DNS (*Domain Name Server*) pelo atacante, que por sua vez realiza o redirecionamento a sites falsos [Ye *et al.* 2008].

A figura 2.2 mostra quais são os possíveis caminhos que um *phiser* utiliza em seus ataques.

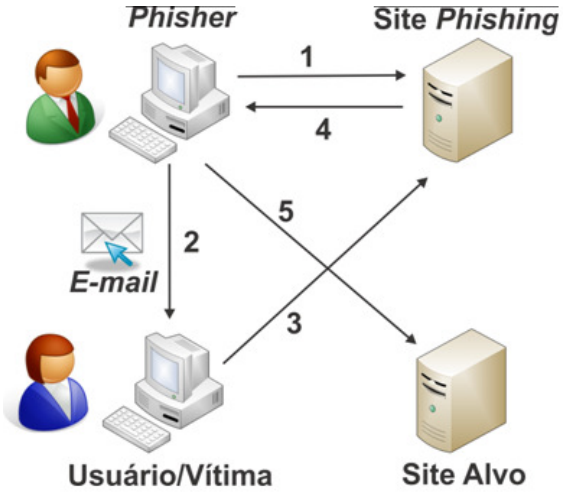

Figura 2.2: Etapas de um ataque *phishing*.

2.3 Técnicas de Prevenção e Detecção

Diversas técnicas têm sido empregadas com objetivo de evitar os ataques *phishing*. Essas técnicas têm sido empregadas juntas ou separadas e contam com diversos mecanismos que serão descritos logo a seguir:

2.3.1 Listas

No cenário comum, as empresas de *anti-phishing* precisam de uma estratégia de detecção precisa e oportuna. As técnicas mais populares implantadas por eles são baseadas em listas negras. Isto é, após a classificação utilizam uma abordagem heurística baseada em recursos ou palavras-chave e, por vezes, os fornecedores precisam compartilhar os resultados com seu destino para alcançar uma alta eficiência de bloqueio ou mecanismo de aviso. Para isso, é

usado uma lista negra que detém compilado URLs, as quais são marcadas como maliciosas.

As *blacklits* são abrodagens baseadas em listas negras (literalmente). Algumas instituições guardam listas de sites fraudulentos e as disponibilizam a quem desejar.Uma lista negra no contexto de *phishing* é uma lista de URLs não confiváveis, ou seja, uma *blacklist* contém sites proibidos que são conhececido por terem inteções maliciosas.

Os navegadores da Internet ocupam um papel importante na interação entre a Internet e os usuários do terminal. A maioria dos fornecedores de segurança de navegador implantaram um método de detecção baseado em *blacklist* integrada diretamente no próprio navegador e barras de ferramentas da web para prevenir usuários de cairem em sites phishing.

Os navegadores mais populares no mercado, como Microsoft IE, Mozilla Firefox, Google Chrome e Opera já lançaram seu navegador integrado com soluções anti-phishing. Nesse particular, especificamente a Microsoft apresentou o seu método baseado em *blacklist* e Microsoft Phishing Filter, em seu navegador Internet Explorer 7, MSN Search Toolbar e Windows Live Toolbar.

O Internet Explorer 9, versão mais recente, utilizou uma estratégia mais poderosa chamada Filtro do SmartScreen, que incluem também a *blacklist* website malicioso. O navegador consulta uma *blacklist* e lista branca (*whitelist*) do servidor para determinar se uma URL é ou não um site *phishing*. O Google publicou um API de Navegação seguro com base na sua *blacklist* atualizada constantemente por sites suspeitos de *phishing*, que é gerado pelo classificador em larga escala [Whittaker *et al.* 2011]. O Google Chrome e Mozilla Firefox estão usando essa fonte, enquanto que outros *anti-phishing* aplicações podem seguir essa API para implantar suas pesquisas ao lado do cliente a partir de uma lista pré existente.

Outras soluções como NetCraft barra de ferramentas, barra de ferramentas Cloudmark, McAfee SiteAdvisor e Norton 360 implantaram sua lista negra privada geradando suas próprias classificações para proteger os usuários do terminal.

Conforme Abu-Nimeh *et al.* [2007], as barras de ferramentas destinadas à detecção de *phishing* estão disponíveis, mas normalmente são utilizadas por usuários sem muitas experiências. Essas ferramentas também são conhecidas como *toolsbar,* que ajudam a suavizar o problema de *phishing.*

Em outra abordagem, realizada por Likarish [2008], é mencionada a utilização de *Whitelist e Blacklist* nas ferramentas *anti-phishing*. Dessa forma, o conteúdo da *whitelist* serão as URLs de sites legítimos, e da *blacklist* dos sites ilegítimos. Muitos dos *anti-phishing*

dependem da combinação entre *whitelist e blacklist.* Assim, o *Google* mantém a combinação w*hitelist e blacklist* para a identificação de *sites phishing.*

Um problema apresentado nessa categoria ocorre quando o usuário entra no site *phishing* por meio da barra de endereço do navegador, pois mesmo que a ferramenta envie um alerta para o usuário, este pode ignorar o alerta e, ao entrar na página poderá estar vulnerável a um *keylogger,* por exemplo.

Outra situação é que esta abordagem está limitada a reconhecer apenas sites que já foram denunciados como *phishing.* Estas listas são alimentadas, em sua maioria, por usuários que serão vítimas ou por usuários já experientes que suspeitam de páginas ilegítimas e as denunciam de modo a serem avaliadas.

2.3.2 Heurísticas

Outro método de identificação, de acordo com Chandrasekaran *et al*. [2006], é o uso de heurística, identificada no processo de comparação das características extraídas de um *site*. Essa comparação é realizada em *sites* legítimos e *phishing.*

A técnica aplicada por Chandrasekaran *et al*. [2006] para classificar páginas *phishing* com base nas propriedades estruturais de *e-mail phishing*, permitiu analisar um total de 25 características de *sites*. Entre essas características existem, por exemplo: a estrutura da linha do assunto e, a estrutura da saudação do corpo do *e-mail*, as quais podem ser traços que quando não identificados corretamente impedem o reconhecimento de um *site* em análise.

A utilização de heurística está sendo aplicada em grandes *sites*, como o Google que utiliza classificadores para identificar *sites phishing* e *sites* legítimos, esse processo é realizado para alimentar sua *blacklist*. [Whittaker *et al.* 2010].

Em abordagens que utilizam heurísticas, o sistema detecta os sites fraudulentos com base em características que existem nos sites *phishing*. Estas características podem ser baseadas na URL. Nesse contexto, o navegador Internet Explore 7, por exemplo, oferece um classificador built-in que filtra as páginas da *Web* com base em sua características.

Os sistemas de detecção automática tentam prevenir o usuário de enviar suas informações a estes agentes mal intencionados. Estes sistemas são geralmente adicionados ao *browser* como um *plugin*, uma extensão ou barra de ferramenta.

Existem diversas ferramentas desenvolvidas para detectar automaticamente uma tentativa de roubo de informações.

O *SpoofGuard* ,por exemplo, é adicionado ao *browser* como uma barra de ferramenta.

Ele é baseado em heurísticas e calcula uma pontuação para a página *web* atual e depois converte esse número em um sinal de trânsito: a cor vermelha indica que o site é um *phishing,* a cor verde que o site é provavelmente seguro [Whittaker *et al.* 2010].

Outra ferramenta que utiliza heurística e destacado como SpoofStick que exibe para o usuário as informações verdadeiras de domínio [Bergholz, A *et al*. 2008]. Um site *phishing* tenta enganar o usuário disfarçando o conteúdo da URL, na tentativa de parecer ao máximo com ao site real. Por exemplo, no endereço www.facebook.com.www,sitefalso.com o usuário poderia achar que está acessando o endereço do facebook, mas na realidade, ele está acessando o site falso – www.sitefalso.com.

Os métodos citados têm vantagens e desvantagens: *blacklist* tem um alto nível de precisão, no entanto, não consegue detectar novos ataques *phishing* imediatamente, requisitando antes a atualização. Ainda, a *blacklist* normalmente requer intervenção e verificação, o que pode acarretar um grande consumo de recursos material e humano. O emprego de heurísticas, por outro lado, consegue detectar *phishing* quando são disseminados, embora possam produzir falsos positivos, ou seja, páginas legítimas incorretamente classificadas como *phishing*.

2.3.3 Educação do usuário

A educação dos usuários é frequentemente recomendada e amplamente utilizada na detecção de *phishing* [Timko, 2008], mas poucos estudos têm avaliado a eficácia das abordagens no mundo real [Kumaraguru *et al.* 2008].

De acordo com Zhang *et al.* [2011], algumas abordagens têm sido concentradas. A primeira pode ser relatada na oferta de informações *online* contra os riscos e ataques de *phishing* e como evitá-los, esses materiais são normalmente fornecidos pelos governos ou organizações sem fins lucrativos. Em outra abordagem são enfatizados testes *online* de habilidades do usuário que atendam a certas pontuações de acertos às páginas ilegítimas, como exemplo cita-se: *Anti-Phishing Phil*, o qual pode treinar os usuários para identificar sites fraudulentos. Os usuários podem aprender como analisar um site para entender a origem do *link*.

PhishGuru motiva os usuários a prestar atenção nos sites que estão acessando ou que ainda irão acessar. Aos usuários do *PhishGuru* são enviados simulados de ataques de *phishing* via e-mail e são apresentados os matérias de treinamento quando os usuários podem evitar cair nos ataques.

Destaca-se ainda que o *PhishGuru* possui uma abordagem de treinamento, ou seja, apresenta procedimentos que orientam os usuários quando estes caem em uma armadilha (*e-mail phishing*). Essa abordagem é importante pela motivação constante quanto ao aprendizado dos usuários.

Existem outras abordagens para treinamento de usuários sobre prevenção de *sites phishing*, incluindo: artigos sobre *phishing* em *sites*, *cartoons online* sobre segurança, avisos de segurança por e-mail e treinamento em sala de aula [Kumaraguru *et al.* 2008].

A análise realizada no trabalho desenvolvido por [Kumaraguru *et al.* 2008] mostra que estas abordagens tem um custo variado e também sua eficácia. Por exemplo: o treinamento em sala de aula pode ser mais eficaz que os outros métodos de formação devido os usuários serem obrigados a passar um tempo dedicado ao treinamento. Contudo, é um processo que pode ser demorado e ter um custo elevado para a empresa. Materiais *online* podem ter baixo custo, masas pessoas dificilmente se interessariam em ler um material extenso, portanto, nem sempre são eficazes.

Os autores Miller e Wu [2005] chegaram a afirmar que os *phishers* exploraram a diferença entre o modelo do sistema e o modelo mental dos usuários com finalidade de enganá-los. Psicólogos e pesquisadores de comunicação têm estudado formas preventivas para evitar que os usuários sejam enganados. Nesse contexto, uma das metas de *anti-phishing* é desenvolver ferramentas de formação dos usuários de modo que eles sejam capazes de gerar e testar hipóteses de ameaças adequadas e protegerem-se de possíveis armadilhas virtuais.

2.4 Aprendizagem de Máquina

De acordo com [Rezende, 2005], aprendizagem de máquina é uma área de IA cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de forma automática.

Nesse contexto, é caracterizado como um sistema de aprendizagem de máquina aquele que é capaz de adquirir conhecimento de forma automática, assim um computador que toma decisões baseado em experiência acumuladas por meio de soluções bem sucedidas de problemas anteriores é sem dúvida uma forma de descrever o conceito de aprendizagem de máquina.

Técnicas de aprendizagem de máquina podem ser utilizadas para o aperfeiçoamento de

determinadas atividades computacionais. Seu uso vai desde o auxílio em diagnósticos médicos até o reconhecimento de escrita, robótica, segurança da informação, etc. [Alpaydin, 2010].

Aprendizagem de máquina, também chamada de *machine learning,* é um termo que engloba um conjunto de metodologias e comportamentos em dados que representam exemplos de acontecimentos do mundo real ou experiências passadas. Dessa forma, os dois objetivos de qualquer projeto de aprendizagem são: induzir o modelo processando uma grande quantidade de dados e realizar inferências a partir dele. Dentre esses objetivos, processar essa grande quantidade de dados é a que exige maior tempo e esforço computacional.

O aprendizado indutivo é efetuado a partir do raciocínio sobre exemplos fornecidos por um processo externo ao sistema de aprendizado. O aprendizado indutivo pode ser dividido em dois tipos principais: supervisionado e não-supervisionado.

No paradigma supervisionado o conhecimento prévio do ambiente é exigido, nesse tipo de paradigma, o objetivo relevante é entender o mapeamento principal da entrada para saídas de informações, nesse projeto, esse será paradigma utilizado.

No aprendizado não-supervisionado, para cada exemplo apenas os atributos de entrada estão disponíveis. O indutor analisa os exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados de alguma maneira, formando os *clusters* [Cheeseman e Stustz, 1990]. Na Figura 2.4 é mostrada a hierarquia do aprendizado descrito acima:

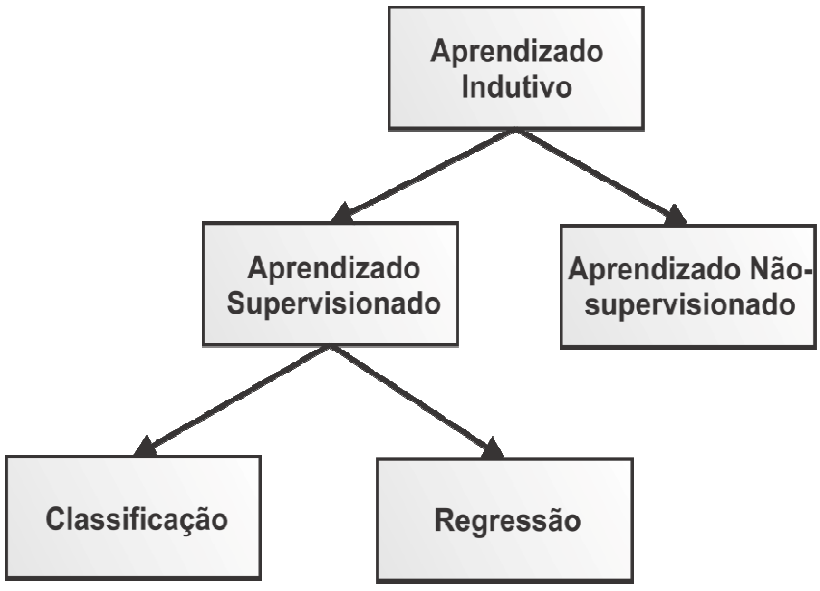

Figura 2.3: Hierarquia do aprendizado

Os algoritmos de aprendizagem de máquina que serão discutidos nas próximas sessões são supervisionados, isto são algoritmos (classificador) que tentam mapear entradas desejando uma saída com uma função específica. No caso de classificação de sites *phishing*, um classificador irá tentar classificar um site como *phishing* ou legítimo, por aprender certas características dos sites.

Nesse sentido, o conceito *aprendizagem de máquina*, em particular classificação automática, tornou-se popular envolvendo trabalhos relacionados à detecção de *e-mail*, s*pam* e detecção de *sites phishing* [Bergholz et. al, 2008].

Os autores afirmam, contudo, que em comparação com os filtros construídos manualmente as regras automáticas avaliam a relevância da entrada de características x $(x_1, \dots x_m)$ (e.g, características de páginas *phishing*) e a estabilidade como função para determinar a classificação desejada de y (e.g, *phishing* ou não *phishing*)

$$y = f(x, \gamma) \tag{2.1}$$

O vetor dos valores de parâmetros desconhecidos é determinado na fase de treinamento, de tal maneira que a relação entre x e y nos dados observados $(x_1,\dots y_1),\dots,(x_D,\dots y_D)$ é reproduzida de acordo com algum critério de otimização. Na fase da aplicação as características iguais são extraídas a partir de uma entrada de novo site. Baseado nessas características o modelo do classificador produz uma classificação de *phishing* e não *phishing*. Para melhor entendimento, a abordagem de aprendizagem de máquina será sumarizada na Figura 2.4.

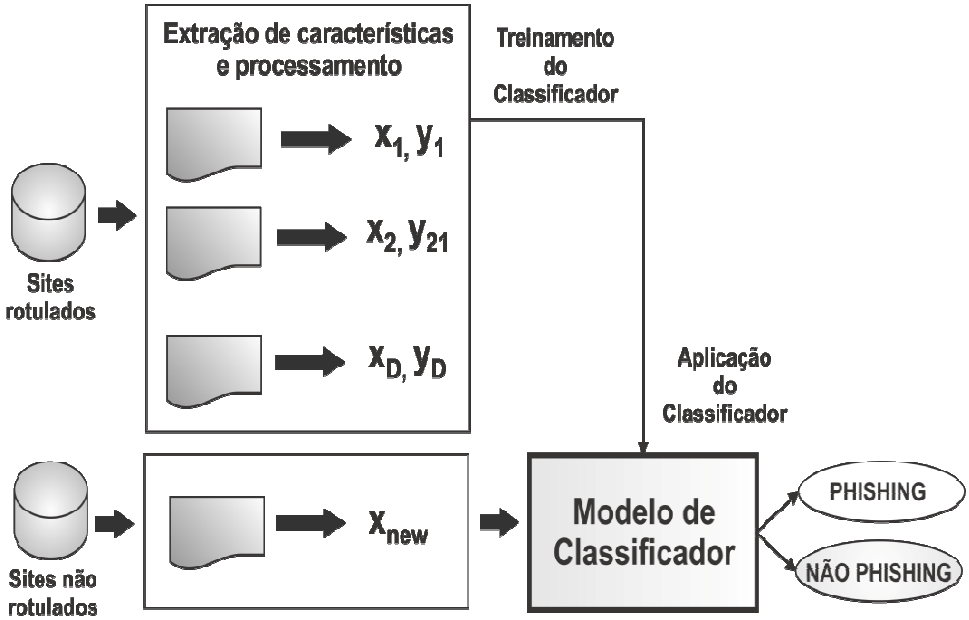

Figura 2.4: Abordagem sobre aprendizagem de máquina.

2.4.1 *Support Vector Machines* (SVM)

Supoort Vector Machine (SVM) é um dos classificadores mais populares hoje em dia. A idéia geral desse classificador é definida ao em encontrar a separação ótima do hiperplano entre duas classes por maximização da margem entre as classes com pontos mais próximos [Abu-Nime et al, 2007].

Nesse contexto, supondo que temos uma função discriminante linear e duas classes linearmente separáveis com valores $+1$ e -1, um hiperplano discriminante pode ser definido como:

$$w'x_i + w_0 \geq 0 \; i + \; t_i = \; +1 \tag{2.4}$$

$$w'x_i + w_0 < 0 \; i + \; t_i = \; -1 \tag{2.5}$$

A distância de qualquer ponto x para o hiperplano e $|w'x_i + w_0 \,|/ \, \|w\| \; e$ a distância para a origem é $|w_0 \;|/\|w\|$. A Figura 2 mostra que os pontos situados sobre o limites são chamados de vetores de suporte, e no meio da margem é o hiperplanos de separação ótima que maximiza a margem de separação [apud Abu-Nime et al, 2007].

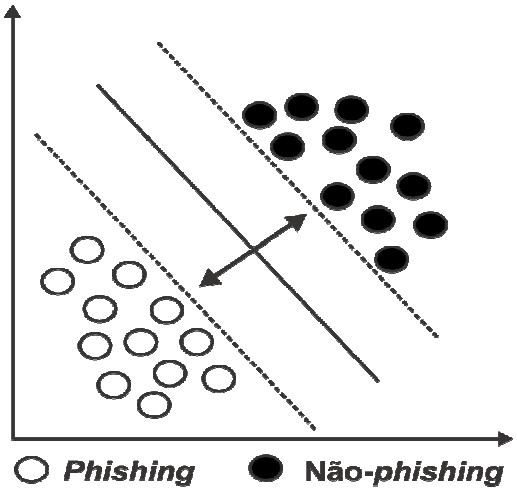

Figura 2.5: Separação linear com SVM.

O problema de detecção de *phishing* pode ser visto como um problema de classificação de duas classes (*phishing* e não *phishing*). Assim, a utilização do algoritmo de aprendizagem de máquina SVM (é adequado, pois foi desenvolvido originalmente para resolver esse tipo de problema [Vapnik 1995]. A literatura técnica mostra que o SVM tem sido

aplicado com bastante sucesso em diversos domínios de aplicação, inclusive na detecção de *phishing* [Basnet, Mukkamala e Sung 2008] e [Chandrasekaran, Narayanan e Upadhyaya 2006].

2.4.2 Naive Baye

O classificador Naive Baye e tratado como um dos algoritmos mais simples, porem eficaz e tem sido utilizado em diversas aplicações, tais como detecção de filtragem de email spam *phishing.* [Miyamoto *et al*., 2009].

Ainda o autor Ye *et al.* [2008] afirma que esse classificar possui uma abordagem mais eficaz tratando-se de classificação de documento de texto. E que dada uma quantidade de amostras de treinamento, uma aplicação pode aprender a partir destas amostras, de maneira que venha a prever a classe da amostra. Ainda, pode-se destacar que um classificador Naive é usado na filtragem de spam, como já mencionado acima e cada e-mail pode ser representado por um vetor com as características : $x = (x_1, x_2, x_3, \ldots x_n)$, onde cada propriedade, $x = (x_1, x_2, x_3, \ldots, x_n)$ é independente.

2.4.3 Árvore de Decisão

Algoritmos que induzem Árvores de Decisão pertencem á família de algoritmos Top Down *Inductions nos Decision Trees (TDIDT).* Uma arvore de decisão (AD) é uma estrutura de dados definida recursivamente como: [Rezende, 2005].

- um nó folha que corresponde a uma classe ou
- um nó de decisão que contém um teste sobre algum atributo. Para cada resultado do teste existe uma aresta para uma subárvore. Cada subárvore tem a mesma estrutura que a árvore.

O processo de construção de uma árvore de decisão pode ser resumido como:

1.Inicia-se com o nó raiz

2. A partir do nó raíz os dados são divididos (uma parte separada dos dados é atribuída a cada nó filho).

3. Cada sub-parte é separada em novas sub-partes até que um critério de parada seja alcançado.

Alguns algoritmos foram desenvolvidos a fim de assegurar a construção de árvores de decisão e seu uso para a tarefa de classificação. O ID3 e C4.5, algoritmos desenvolvidos por Quinlan [1993], são provavelmente os mais populares.

Comentários finais: Foi apresentado uma breve ilustração no tema aprendizagem de máquina e os algoritmos utilizados neste trabalho, para um melhor aprofundamento dos assuntos relacionados acima, os seguintes livros podem ser consultados: Alpaydim, E. (2004) "Introduction to Machine Learning" The MIT Press. Cambridge, Massachusetts, EUA. 415 pp e Witten, Ian; Eibe, Frank; Hall, Mark A. (2001). Data Maning "Practical machine learning tools and techniques. 3º Ed. Elsevier. USA.

Capítulo 3

Trabalhos relacionados e suas classificações

Neste capítulo são apresentados alguns conceitos e definições preliminares que serão tratados no decorrer desta dissertação, os quais são fundamentais para a compreensão temática investigativa. Também são apresentados alguns trabalhos relacionados que servem como referência na detecção de páginas *phishing*.

3.1 Visão Geral

Conforme Zhang *et al*. [2007], são lançadas uma variedade de barras de ferramentas *anti-phishing*. Para melhor exemplificação foi identificado o *site ebay,* o qual lançou uma ferramenta grátis que reconhece positivamente seu próprio *site*.

Fette *et al*. [2007] afirmam que as primeiras tentativas de aplicações de aprendizagem de máquina para problema de *phishing* foram nas *toolbars,* tais como *Spoofguard* [2001] e Netcraft [2006]. Um problema apresentado nessa categoria, conforme *Fette et al*. [2007], é que as *toolbars* possuem acesso a uma quantidade pequena de informação.

Na solução proposta por Likarish *et al.* [2008], B-APT1 (*Bayesian Anti-Phishing Toolbars*), que utiliza um filtro *bayesian*, ao aplicar em *toolbars* tem saldo positivo na perspectiva do autor, pois a técnica foi mais bem sucedida do que os "filtros baseado em regras". Ainda, no ponto de vista de Likarish *et al*. [2008], também é enfatizado que os filtros *baysianos* efetivamente podem marcar *e-mail spam* nunca visto anteriormente.

Dessa forma, a B-APT detectou 100% dos *sites phishing*, enquanto SpofGuard [2010] e Netcraft (2006) detectaram 88% e 63% respectivamente, o que levou a B-APT a atingir a melhor taxa de acerto, sendo considerada a precursora na utilização das redes *bayesianas* em sua *toolbar.*

3.2 Listas: Blacklist e Whitelist

A abordagem de Likarish *et al*. [2008] destaca que muitos *anti-phishing* dependem da combinação entre *blacklist* e *whitelist* para classificar uma página verdadeira ou ilegítima. O *Internet Explore* 7, por exemplo, bloqueia o acesso do usuário consultando sua *blacklist*.

Na perspectiva de Sheng *et al*. [2009], a preparação e distribuição da *blacklist* são realizadas considerando vários processos, entre estes a inclusão de *e-mails* que foram gerados de *spam traps* ou detectados por filtro *spam* sugeridos por usuários (i.e Phishtank e APWG).

Nesta direção, para Sheng *et al*. [2009], a atualização das *blacklist* é ponto fundamental no âmbito dessa discussão, uma vez que o tempo em que a página *phishing* permanecer no ar sua inclusão na *blacklist* não poderá exceder a 30 minutos. A utilização de heurísticas nesse processo ocorre com a utilização de *blacklist* e heurísticas em conjunto.

Por tudo isso, Ye *et al.* [2008] apontam que a maioria das detecções de *phishing* ocorre baseada em *blacklist,* pois esta é responsável por acionar alerta do ataque quando um usuário visita um site que esteja em seu banco de dados. Portanto, manter a *blacklist* atualizada exige grande dose de esforço, considerando que a todo momento surge novos *sites phishing*.

Ainda dentro da proposta de Ye *et al*. [2008] é apresentada uma ferramenta intitulada *Automated Individual White-List* (AIWL), esta arquiva o *Login User Interfaces* (LUIs) em uma *whitelist* utilizando o classificador *Naive Bayes*, o qual tem o papel de identificar se o *login* do usuário de fato foi realizado com sucesso. Neste caso, as vantagens apontadas nesta proposta é que AIWL reconhece sites *pharming* verificando o endereço IP do site, principalmente levando em conta que os sites populares possuem endereço estável, de modo que AIWL pode detectar *pharming*.

Com a utilização do classificador *Naive Bayes,* a AWIL teve uma taxa de falso negativo de 100% e, falso positivo de 0% na identificação do processo de *login* do usuário, resultado satisfatório que se baseia no comportamento do *login* atual do usuário em um determinado *site* [Ye *et al*., 2008].

A eficiência da *whitelist* é devido o conteúdo inserido nela ser estável. Em contrapartida, caso AIWL seja instalado numa máquina local é dificil controlar ataques do tipo cavalo de troia e vírus de computador, sendo necessário armazenar essa *whitelist* em um telefone inteligente.

Segundo Ma *et al*. [2009], informar aos usuários sobre sites fraudulentos significa que parte

do problema referente aos ataques *phishing* seria atenuada. Para isso, as *blacklist* sugeridas para fornecer aos usuários a notificação de *sites* fraudulentos estão embutidas na *toolbar* do navegador.

Segundo Ma *et al*. [2009], a criação de *blacklist* é realizada por uma série de técnicas, como: relatório manual, *honeypots*[1] e coletor *web (crawlers)*[2], combinando com site de análise heurística. A utilização de aprendizagem de máquina (AM) faz-se necessário nesse processo realizando a classificação da reputação dos *sites* entre URLs e as características lexicais dos *hosts.*

De acordo com Ma *et al*. [2009], a confecção do conjunto de dados contou com 20.000 *sites*, sendo 15.000 do tipo legítimo, 5.500 do tipo *phishing* e 15.000 *spams*. Diversos experimentos foram realizados com combinações entre sites do repositório PhishTank e Dmoz. A utilização de classificadores *Naives Bayes*, *Suporte Vetor Machine* (SVM) e Regressão Logística (LR) compõem esse processo, e a análise do impacto de cada uma das características foi muito satisfatória. Com a utilização de um conjunto de dados do Yahoo e PhishTank, as taxas atingidas para falso positivo e falso negativo foram respectivamente: 0,1% e 7,6%, usando Regressão Logística.

Dessa maneira, para Sheng *et al*. [2009], a prevenção e a detecção de *phishing* podem ser feitas em dois níveis de abordagens: filtragem de *emails* e de páginas *Web*. A maioria dos navegadores integra como recurso de proteção contra *phishing* as barras de ferramentas que utilizam *blacklist,* e algumas vezes heurísticas para detectar *sites phishing*. O favorecimento atribuído a *blacklist* resulta da apresentação baixa da taxa de falso positivo, em relação à heurística. Os autores, contudo, destacam a utilização de *blacklist* e heurística para maior apanhado de *sites phishing*.

Um conjunto de ações para verificar a duração de um *site phishing* foi analisado nesse trabalho, os *sites phishing* utilizados foram os da University of Alabama (UAB), sendo coletados e testados 191 *URLs* de *phishing.*

O levantamento que determinou a duração do tempo de vida de s*ite phishing* culminou com os seguintes resultados: de 191 URLs, 127 URLs, ou seja, 66% obtiveram uma duração de menos de 24 horas e, 25 URLs obtiveram duração entre 24 e 48 horas, sendo o restante das

[1] Software, cuja função é detectar ou de impedir a ação de um cracker, de um spammer, ou de qualquer agente externo estranho ao sistema, enganando-o, fazendo-o pensar que esteja de fato explorando uma vulnerabilidade daquele sistema.
[2] Programa de computador que navega pela *World Wide Web* (WWW) de uma forma metódica e automatizada.

URLs entre 24 horas a 23 dias. É importante salientar o tempo de vida de um *site phishing* porque a *blacklist* baixa site *phishing* em tempo geralmente muito lento [Sheng *et al*., 2009].

A base de dados para detecção de falsos positivos contou com 13.458 *sites* legítimos e 10.000 de *sites phishing*. Contudo, a taxa de falso positivo se equipara a quase 0% para *phishing* contido em *blacklist.*

A seguir uma abordagem sobre ferramentas *anti-phishing* que utilizam heurísticas no processo de detecção de *phishing* enfatizando métodos e métricas específicas em Aprendizagem de Máquina (AM).

3.3 Heurísticas

As heurísticas procuram verificar se a página supostamente legítima possui características de um ataque *phishing*. Conforme Sheng *et al*. [2009], a maioria das heurísticas para detectar sites *phishing* usa as seguintes características: conteúdo da HTML, conteúdo dos sites ou assinaturas URL para identificar *phishing*. Os algoritmos de aprendizagem de máquina são normalmente aplicados para construir modelos de classificação sobre as heurísticas, de maneira a classificar novas páginas *web*.

No trabalho realizado por Abu-Nimeh *et al*. [2007], a detecção de *e-mails phishing* se processa por meio das características extraídas de corpo do *e-mail,* apresentando as mais recentes tendências de *e-mail phishing*. O conjunto de dados das páginas analisadas pelo autor consistiu na identificação de 43 características, as quais apresentam resposta binária como *phishing* =1 e, legítimos = 0. Além disso, o autor utilizou TF-IDF (Termo freqüência-inversa do original do termo), que é o algoritmo usado em recuperação da informação desenvolvido por [Phelps *et al*., 2000].

No contexto tratado acima foram aplicados os classificadores: Regressão Logística (RL), *Classification and Regressions Trees* (CART), *Bayesian Additive Regression Tress* (BART), SVM, Floresta Aleatória (FA) e *Neural Network* (NNET), considerando que todos esses classificadores realizaram a estratégia de validação cruzada em dez partes. A base de páginas é dividida em dez partes iguais e mutuamente exclusivas entre si. Cada parte é testada, sendo nove partes usadas como base de treino do classificador. Esse processo é realizado para cada uma das dez partes cruzadas e, ao final é calculada a média da classificação obtida nas dez interações.

Os resultados obtidos no experimento de Abu-Nimeh *et al.* [2007] apresenta detecção

de sites *phishing* e legítimo com pesos iguais, e o classificador FA tem a melhor taxa de erro, equivalente a 7,72%. Assim sendo, o classificador mencionado alcança o pior índice de falso positivo, o equivalente a 8,29%. Quando aplicado o custo de erro, penalizando falsos positivos por nove vezes a mais que falsos negativos, a taxa do classificador RL supera a taxa de todos os classificadores, alcançando taxa de erro de 3,82%. A pior taxa quando aplicada, a penalidade é de FA com 5,79%, assim, para detecção de email, seja *phishing* ou *spam*, requer a aplicação de penalidade em falsos positivos, estes possuem um custo maior no mundo real.

A utilização de aprendizagem de máquina aplicando um classificador automático para detecção de páginas *phishing* alimenta automaticamente a *blacklist,* o que é proposto por [Whittaker *et al*. 2010]. A idéia consiste em realizar a análise de milhões de páginas diariamente por esse projeto, encontrando entre as características analisadas: URL e o conteúdo de uma página para determinar *phishing*. A diferença dos demais projetos desse trabalho é que a data set é processada contendo ruídos.

Ainda assim, para a montagem da base de dados o sistema classifica páginas *web* submetidas pelos utilizadores finais e recolhidas de filtros de *Spam* e do *Gmail*, com essas páginas é extraída uma série de características, estas descrevem a composição de URL da página, hospedagem da própria página e da HTML do conteúdo.

A classificação com Regressão Logística se faz presente nesse processo contando com a vantagem que menos de 1% da entrada é realizada manualmente. O algoritmo de recuperação de informação TF-IDF também compõe esse sistema que mantém uma taxa de falsos positivos abaixo de 0,1%.

Esse projeto utiliza a combinação de vários métodos para detecção de *phishing* que apresenta taxas baixas na utilização de heurísticas, e no momento de classificar a inclusão da página na *blacklist* medições são realizadas com o objetivo de classificar corretamente.

A proposta lançada por Zhang, Yue *et al*. [2007] é a implementação de Cantina (*Carnegie Mellon Anti-phishing and Network Analysis Tool*), um novo contexto baseado em aproximação para detecção de páginas *phishing.* A Cantina examina o conteúdo da página, por exemplo, a URL, e o nome do domínio utilizando o algoritmo TF-IDF.

A ferramenta Cantina pode detectar de 94 a 97% de *sites phishing*, mostrando que é possível utilizar um conjunto de heurísticas com outras ferramentas para reduzir o falso positivo. Com a utilização do algoritmo TF-IDF é possível detectar cerca de 97% sites *phishing* com apenas 6% de falso positivos e, combinando algumas heurísticas a taxa é de

90% de sites *phishing*, com apenas 1% de falsos positivos.

Apesar da utilização de técnicas mescladas no emprego de aprendizagem de máquina e algoritmo de recuperação de informação, a taxa de falsos positivos se equipara [Whittaker, 2010]. Porém, a taxa de detecção entre 94 e 97% deixa em torno de 6% os sites classificados indevidamente como legítimos.

Como salientado por Miayamoto *et al*. [2009], a detecção da precisão em soluções baseadas em heurísticas estão longe de serem ideais, pois para melhorar a precisão utilizaram heurísticas com aprimoramento do cálculo da probabilidade. Nesse sentido, para melhorar a detecção da previsão, os autores empregaram *AdaBoost*, uma técnica de AM, como método para calcular a probabilidade da previsão, e assim apresentar resultados melhores.

Os autores utilizaram técnicas de aprendizagem de máquina, tais como: *AdaBoost*, *Bagging*, SVM, CART, RL, Floresta Aleatória, Redes Neurais (NN), *Naive Bayes* (NB) e BART. A base de dados foi composta baseada no critério estabelecido por CANTINA [Zhang, *et al*. 2007].

Nesta base de dados deve haver a mesma quantidade de *sites phishing* e legítimo, respectivamente. Nesse sentido, a base de dados foi composta por 1.500 de *sites phishing* e 1.500 de *sites* legítimos, obtendo taxa de falso positivo de 13,64%, com NB, sendo a menor taxa de falso negativo 13,54 com NN.

Nessa direção, Fette *et al*. [2007] propõem um método intitulado PILFER (do Inglês *Phishing Identification by Learning on Features of Email Received*) que é um algoritmo que identifica *phishing* por meio de AM sobre as características de e-mail recebido.

Um conjunto de dados foi utilizado para treinar e testar a base de dados usando validação cruzada com dez (10) partições para melhor obter a média do resultado, tendo sido utilizado o classificador SVM. Para montagem do conjunto de dados, dois conjuntos de dados disponíveis forma utilizados: *corpora ham (SpamAssassin*) - 6.950 amostras com *e-mails* legítimos e, 860 *e-mails phishing* retirados do *phishincorpus*.

O conjunto de dado PILFER conseguiu um total de 99% de precisão, uma taxa de falso positivo de menos de 1%. Por outro lado, uma taxa de falso negativo de 7 a 8%, o que equivale à metade do *SpamAssassin*.

Nesse sentido, Fette *et al.* [2007] afirmam que é possível detectar *e-mail phishing* com alta precisão usando filtro especializado, assim como a utilização de recursos que são

diretamente aplicados nos e-mails *phishing,* diferentemente dos empregados por filtros *spam* de propósito geral. A utilização do algoritmo PILFER remove a interação do usuário, deixando este sem chance de dispensar diálogos de alertas, além de apresentar taxas de precisão de falso positivo baixo.

Soluções apresentadas nesta seção para detecção de *phishing* são destacadas na Tabela 3.1 a seguir:

Tabela 3.1 Soluções para detecção de *phishing.*

Publicações	**Métodos e Técnicas**	**Resultados em AM**
Fett, Ian *et al*., (2007)	Floresta Aleatória	Precisão: 99%, FN: 7% e FP: 1%
Likarish,Peter et al., (2008)	Filtro *Bayesian*	100% de acertos em sites *phishing*
Ye, Cao et al., (2009)	*Whitelist, Naive Bayes* (Processo de *Login*)	FN: 100%, FP:0% (no processo de *login*)
Ma, Justin et al., (2009)	*Blacklist* e Regressão Logística	FP: 0,1 e FN: 7,6%
Sheng, Steve, et al. (2009)	*Blacklist* e Heurística	FP: 0% (*nas blacklists*)
Abu-Nimeh et al., (2007)	TF-IDF e Regressão Logística	FP: 3,82%
Whittaker, Colin et al., (2010)	*Blacklits*, Regressão Logística e TF-IDF	FP: 0,1%
Zhang, Yue et al., (2007)	AM, TF-IDF	Acerto: 97% e FP: 6%
Miayamoto et al,. (2009)	*Naive Bayes* e Redes Neurais	Precisão: 99%, FN: 7% e FP: 1%

Legenda: FN (Falso Negativo), FP (Falso Positivo) e AM (Aprendizagem de Máquina).

Capítulo 4

Metodologia Utilizada no trabalho proposto

Neste capitulo será apresentado a metodologia utilizada e a descrição das etapas compostas, de forma a explicitar cada passo realizado no experimento.

4.1 Modelagem da base de dados

A metodologia adotada neste trabalho está dividida em cinco etapas: download da página, normalização, extração das características, classificação e análise das características.

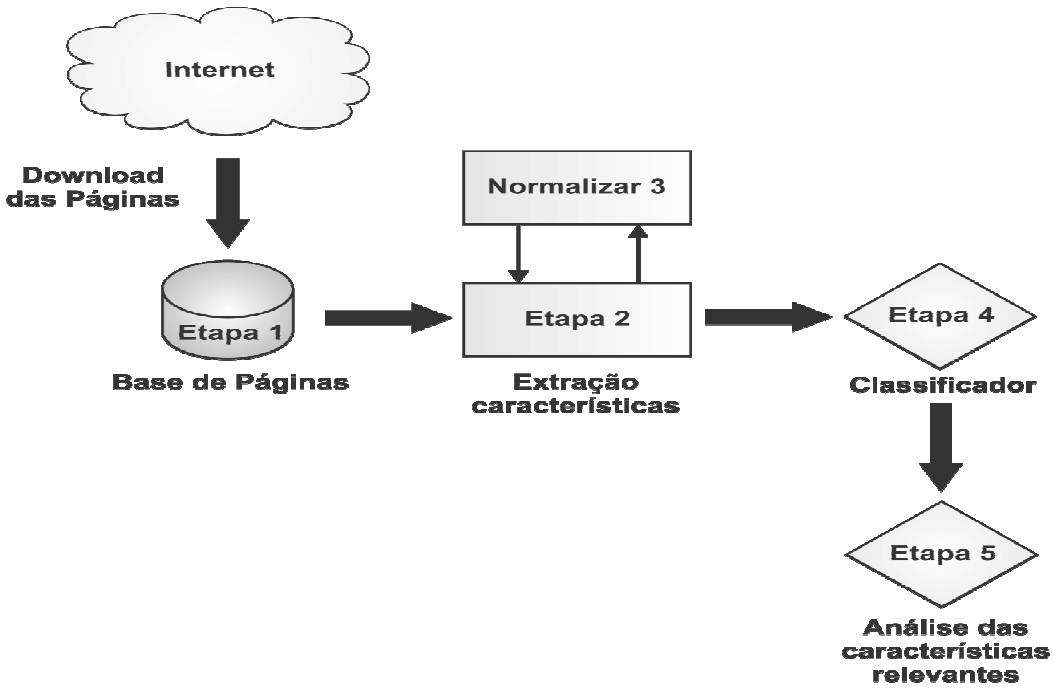

Figura 4.1: Etapas da metodologia adotada.

4.2 Download da página

A análise das páginas pode ser feita tanto em tempo real (*online*), como a partir de arquivos

salvos (*off-line*). No entanto, como páginas *phishing* ficam *online* por um curto período de tempo, em média 72 horas, o *download* é necessário. Isto permite que análises posteriores sejam feitas na página, quando esta já não estiver disponível na Internet.

4.3 Extração das características

A partir dos trabalhos relacionados e da análise das páginas na base de dados, foram selecionadas dezeito características que evidenciam a diferença entre páginas *phishing* e páginas legítimas: URL baseada em IP, quantidade de pontos na URL, tamanho da URL, caracteres suspeitos na URL, domínio de topo, palavras-chave na URL, links para outros domínios, objetos para outros domínios, presença de formulário, palavras-chave no título da página, geolocalização do servidor de hospedagem, *pagerank* da página no Google, caracteres hexadecimais, idade do domínio, anonimizador, algoritmo TF-IFD e velocidade da conexão.

4.3.1 Características extraídas da URL

C1: URL baseada em IP

Algumas páginas *phishing* são hospedadas em máquinas que não têm registro no DNS, a saída para o *phisher* é referenciar pelo IP; isso também previne que o site seja tirado do ar pelo desativamento do domínio. Além disso, entidades legítimas raramente usam IPs em suas URLs.

C2: Quantidade de pontos na URL

Uma das formas de obscurecer a URL e tentar fazer com que os usuários acreditem que a URL seja verdadeira é usar subdomínios, como em http://www.bank.com.br.badsite.com/ - ou utilizar o domínio alvo de *phishing* no caminho, como em http://badsite.com/www.bank.com.br/. Em ambos os casos existe uma grande quantidade de pontos.

C3:Tamanho da URL

Através da análise da base foi verificado que as URLs de páginas *phishing* costumam ter uma quantidade muito maior de caracteres do que as páginas legítimas, numa tentativa de desviar a atenção do usuário.

C4: Palavras-chave na URL

Existem determinadas *strings* que são comuns em URLs de *phishing*, usadas na tentativa de

dar segurança ao usuário: account, update, confirm, verify, secur, notif, log, click, inconveninen, ebay, paypal.

C5: Caracteres suspeitos

A ferramenta SpoofGuard identifica dois caracteres encontrados na URL como suspeitos: o "@" (arroba) e, o "-" (underline). Destes, foi utilizado apenas o arroba, o qual é o mais perigoso uma vez que, quando presente no domínio, tudo o que vem antes dele é considerado nome de usuário e repassado ao endereço da página. Assim, ao acessar http://www.bank.com.br@badsite.com/, o usuário pode pensar que está acessando o site http://www.bank.com.br, mas na realidade está navegando em http://badsite.com, sendo o nome de usuário www.bank.com.br.

C6:Domínio de topo (TLD)

Um nome de domínio é composto por uma série de nomes separados por pontos. O último desses nomes é o chamado domínio de topo que pode ser organizado em dois grupos: TLD - com duas letras que representam países (como ".br" que representa o Brasil), e TLD - com mais de duas letras usados para propósitos genéricos (como ".gov" que é de uso restrito para entidades do governo). Domínios podem ser registrados em alguns desses TLD sem restrições enquanto outros devem seguir alguns pré-requisitos, assim determinados TLD podem oferecer maior facilidade de serem usados por *phishers.*

C7: Anonimizador

Alguns *phishers* adicionam URL idênticas do site original, completando com site onde se encontra a página hospedada, normalmente esse site de hospedagem é grátis como no exemplo: http://www.bb.com.br. v10.com.br. Em casos assim, o *phisher* espera que o usuário não perceba o final do site "v10.com.br" e que "bb.com.br" chame mais atenção do usuário;

C8: Quantidade de subdomínios na URL

Alguns *phishers* adicionam subdomínios para dar uma aparência mais confiável à URL utilizando nomes de entidades autênticas e bem conhecidas como no exemplo abaixo:

http://recadastro.receitafederaldobrasil.badsite.com

Em casos assim, o *phisher* espera que os subdomínios "recadastro" e "receitafederaldobrasil" chamem mais atenção do usuário do que o próprio domínio badsite.com.

C9: Caracteres hexadecimais

O caractere "%", quando lido pelo navegador web, indica que os próximos dois caracteres lidos são hexadecimais. Com este artifício é possível "disfarçar" alguns caracteres. Contudo, conforme Pan & Ding [2006] muitos sites legítimos utilizam notação hexadecimal para representar símbolos de pontuação como aspas, caractere de espaço, ponto de interrogação, etc., assim essa característica só faz sentido quando o valor hexadecimal é uma letra ou número inválido.

4.3.2 Características extraídas a partir de informações em bases de dados *online*

Informações sobre sites são armazenadas por diversas companhias em bases de dados com finalidades específicas, como prover informações de rastreamento e indexação. Essas informações por si não podem comprovar que uma página é *phishing*, mas podem reforçar evidências encontradas em outras características. Duas características desse tipo foram extraídas:

C10: Geolocalização da página

A hospedagem de páginas *phishing* pode se concentrar em determinadas regiões do planeta. Essa característica foi implementada a partir da base de dados fornecida pela empresa MaxMind (2011), onde blocos de IPs estão relacionados aos países.

C11: Google PageRank™

O pagerank é um valor numérico que representa a importância de uma página num conjunto de páginas *web*. Quanto maior o *pagerank* de uma página, mais importante ela é. Mas como páginas *phishings* têm um curto período de vida, seu *pagerank* é muito baixo ou inexistente.

C12: Idade do domínio

Páginas *phishing* normalmente têm um curto período de vida. Os domínios são registrados poucos dias antes dos *e-mails phishing* serem enviados. Nós sinalizamos páginas que foram registradas a menos de 60 dias da data da coleta ou páginas em que essa informação está indisponível. Foi feita uma busca WHOIS para implementar essa característica.

4.3.3 Características extraídas a partir do conteúdo da página

Mesmo que a URL pareça legítima, é possível determinar se uma página é *phishing* analisando o conteúdo da mesma. Para isso, foram extraídas quatro características a partir do HTML da página:

C13: Presença de formulário

Através de um formulário com campos para entradas de texto o *phisher* consegue obter os dados pessoais das vítimas. Caso a tag <INPUT> esteja presente no HTML da página e for do tipo "textfield" ou "password" a página é sinalizada. Páginas legítimas que têm essa característica são facilmente distinguíveis de *phishing* através das outras características.

C14: Razão de *links* para outros domínios

Normalmente os *links* em uma página apontam para o mesmo domínio. Em páginas *phishing*, na tentativa de ficar o mais parecido possível com a página real, os *links* normalmente apontam para o domínio real. Nesta característica é calculada a razão R_1 conforme equação abaixo:

$$R_1 = \frac{L_O}{L} \tag{4.1}$$

onde L_O é a quantidade de *links* para outros domínios e L a quantidade total de *links* na página.

C15: Razão de objetos carregados a partir de outros domínios

Uma página *Web* é composta de diversos objetos incluindo imagens, *css, iframes, scripts*, etc. Em uma página comum, a grande maioria desses objetos é carregada a partir do próprio domínio. Em páginas *phishing* é comum os objetos serem carregados a partir do site real. A razão R_2 entre a quantidade de objetos carregados a partir de outros domínios é calculada pela equação abaixo.

$$R_2 = \frac{O_O}{O} \tag{4.2}$$

Onde "O_O" é a quantidade de objetos para outros domínios e "O" a quantidade total de objetos na página.

C16: Palavras-chave no título do site

Assim, como nas URLs, algumas *strings* são comuns no título de páginas *phishing*. Foi utilizado o mesmo conjunto de *strings* aplicado à extração dessa característica na URL para a extração no título da página.

C17: Algoritmo TF-IDF

È um algoritmo que calcula o quão é importante um termo para um documento. Esse valor é

obtido contando com o número de vezes que o termo aparece no documento, dividido pelo log da frequência desse termo em todos os documentos. Páginas *phishing* usam termos comuns nas páginas de seus alvos e isso reflete num valor alto de TF-IDF para esses termos. Já em páginas normais esses termos não terão um valor alto, pois não são relevantes.

C18: Velocidade da conexão

Se alguns sites maliciosos tende a residir em máquinas comprometidas, tais como as residenciais (conectada via cabo ADSL), então e adequado registra a velocidade de conexão do host.

4.4 Normalização

É o processo formal passo a passo que examina os atributos de uma entidade, com o objetivo de evitar anomalias observadas na inclusão, exclusão e alteração de registros.

Objetivos

- Minimização de redundâncias e inconsistências;
- Facilidade de manipulações do banco de dados;
- Facilidade de manutenção do sistema de Informação.

4.5 Classificação

Como citado na Seção 2.3, um classificador depois de treinado consegue predizer a qual classe uma amostra não rotulada pertence a partir da leitura do seu vetor de características. O uso desses algoritmos torna a detecção de *phishing* muito mais eficiente, onde ao invés de se criar e atualizar manualmente as regras de filtragem de dados, eles o fazem automaticamente. Neste trabalho foram utilizados os classificadores SVM, *Naive Bayes* e Árvore de Decisão.

4.6 Análise da relevância de cada característica

Os algoritmos de aprendizagem de máquina ajudam a identificar quais são os atributos mais adequados a serem utilizados para tomar decisões. Os objetivos com a eliminação dos atributos irrelevantes são:

- Melhorar o desempenho dos algoritmos de AM;
- Simplificar o modelo de predição e reduzir o custo computacional;

- Fornecer um melhor entendimento sobre os resultados.

Tabela 4.1: Sumarização dos modelos de detecção de *phishing*.

Características	Autores Relacionados								
	Fett et al., (2007)	Abu-Nimeh et al., (2007)	Zhang et al., (2007)	Likarish et al., (2008)	Ye et al., (2009)	Ma et al., (2009)	Sheng et al. (2009)	Miayamoto et al,. (2009)	Whittaker et al., (2010)
C1:URL baseada em IP	X		X			X		X	X
C2: Quantidade de pontos na URL	X		X					X	X
C3:Tamanho da URL	X		X			X			X
C4: Palavras-chave na URL	X		X			X			X
C5: Caracteres suspeitos			X					X	X
C6:Domínio de topo (TLD)						X			
C7: Anonimizador						X			X
C8: Quantidade de subdomínios na URL									
C9: Caracteres hexadecimais									X
C10: Geolocalização da página						X			X
C11: Google PageRank™			X						X
C12: Idade do domínio	X		X		X	X		X	X
C13: Presença de formulário			X		X			X	X
C14: Razão de links para outros domínios									
C15: Razão de objetos carregados a partir de outros domínios								X	
C16: Palavras-chave no título do site	X							X	
C17: Algoritmo TF-IDF		X	X					X	X
C18: Velocidade da conexão						X			
Não Documentadas		X		X		4	X(5)		
Total de Caracteristicas	6	2	9	1	2	12	4	8	12
Tamanho da Base de dados									
Phishing	860	1718	100	não mencionada	18	6000-7500	191	1500	não mencionada
Não Phishing	6950	1171	100	não mencionada	16	2 milhões	não mencionada	1500	não mencionada
Base de Dados (Phishing e Não phishing respectivamente)	SpamAssassin e Phishingcorpus	Spambase	PhishTank e 3Sharp's	Phishtrack e Alexa	PhishTank e coletada manualmente	Não mencionada e Yahoo!	Universidade de Alabama	PhishTank, 3Sharp, Alexa Web Search e Yahoo!	Google

Capítulo 5

Experimentos e Análise dos Resultados Preliminares

Nesta seção são apresentados detalhes sobre o ambiente utilizado para a realização dos experimentos, incluindo os parâmetros e as configurações do classificador SVM, Naive Bayes e Árvore de Decisão e filtros *Wrapper*, InfoGain e *CfsSubSet* na análise da relevância das características implementadas. Todos os experimentos foram desenvolvidos em uma estação de trabalho Core i7, 2,67 MHz, com 4GB de memória e um disco SATA com 500 GB de espaço de armazenamento. Os algoritmos para aprendizagem de máquina, filtros de análise das características foram executados através da ferramenta *Weka*.

5.1 Experimentos

5.2 Base de Dados

Como não existe uma base de dados de *phishing* normalizada, ou seja, rotulada e com as características extraídas, foi necessário preparar uma base de cunho próprio. Esta base contém 20.000 páginas, sendo que 10.000 amostras de *phishing* e 10.000 amostras de páginas legítimas. As amostras de *phishing* foram coletadas a partir do repositório *PhishTank* [2011] entre os dias 01/12/2012 a 01/02/2013 amostras de páginas legítimas foram sorteadas da lista composta por mais de 4 milhões de páginas fornecida pelo *Open Directory Project* [2010] e *Clueweb* 2009.

A construção da base de dado adotou critérios para escolher URLs, com base nos critério da CANTINA, foi coletadas URLs com o mesmo número de *phishing* e sites legítimos, conforme ilustrado na figura 5.1.

5.3 Métricas

5.3.1 Desempenho geral

Quatro métricas foram utilizadas para avaliar o desempenho da classificação:

a) Precisão: definida como a razão entre quantidade de páginas *phishing* corretamente classificadas e a quantidade total de páginas classificadas como *phishing*.
b) Taxa de verdadeiros positivos: corresponde à razão entre a quantidade de páginas *phishing* corretamente classificadas e quantidade total de páginas *phishing*.
c) Taxa de falsos positivos: calculada pela razão entre o número de páginas legítimas classificadas como *phishing* e a quantidade total de páginas legítimas.
d) Taxa de acertos: é a razão entre a quantidade de páginas corretamente classificadas e o número total de páginas.

5.4 Resultados

Referências

Anti-Phishing Working Group. (2010) “Phishing Activity Trends ReportQ1 2010”, Disponível em: http://www.antiphishing.org/reports/apwg_report_Q1_2010.pdf.

Abu-Nimeh, S., Nappa, D.; Wang, X.; and Nair, S. (2007) “A Comparison of Machine Learning Techniques for Phishing detection”, In: Proceedings of the Anti-phishing Working Groups 2nd annual eCrime Researchers Summit (eCrime '07), pp. 60-69.

Alpaydim, E. (2004) “Introduction to Machine Learning” The MIT Press. Cambridge, Massachusetts, EUA. 415 pp.

Basnet, Ram ; Mukkamala, Srinivas; Sung, Andrew H.(2008) “Detection of Phishing Attacks: A Machine Learning Approach” , SPRINGER, Verlag Berlin Heidelberg ,pp. 373–383.

Changxin, Gao. Phishing websites rake in $3 billion. China Daily, Shanghai, 15 janeiro 2011. Disponível em:

<http://www.chinadaily.com.cn/china/2011-01/15/content_11859319.htm>. Acesso em: 25 Novembro 2011.

CERT.br – Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil. (2011) *Estatísticas do CERT.br*. Disponível em: <http://www.cert.br/stats/incidentes>. Acesso em: 16 Novembro de 2011.

Código Penal Brasileiro, Título II, Cap. VI, Art. 171, Disponível em: <http://www.planalto.gov.br/ccivil_03/Decreto-Lei/Del2848.htm>. Acesso em 05 Jan. 2012.

Kuo, Cynthia; PARNO, Bryan; PERRIG, Adrian. Browser Enhancements for Preventing Phishing Attacks. Disponível em: <http://research.microsoft.com/pubs/138297/browser.pdf> Acesso em: 12 outubro 2011.

Kumaraguru, Ponnurangam., Sheng, Steve., Acquisti, Alessandro., Cranor Lorrie F., Hong, Jason (2008), “Lessons From a Real World Evaluation of Anti-Phishing Training” In: *IEEE International Conference System* , eCrime Researchers Summit, pp. 1-12.

Downs, Julie S.; Holbook, Mandy B.; Cranor, Lorrie. (2006) “Decision Strategies and Susceptibility to Phishing”. In: *Proceedings of the second symposium on Usable privacy and security* (SOUPS '06). ACM, New York, pp. 79-90.

Fette, Ian; Sadeh, Norman; Tomasic, Anthony. (2007) “Learning to Detect Phishing Emails”. In: International Conference On World Wide Web, ACM, New York, 2007, 16th, n. 1357, pp. 649-656.

Garera, Sujata et al. (2006) “A Framework for Detection and Measurement of Phishing Attacks” In: International Conference On World Wide Web, ACM, New York, 16, pp.1-8.

Loftesness, S. (2004) *Responding to "Phishing" Attacks*.Glenbrook Partners.

Litan, A. (2004) *Phishing Attack Victims Likely Targets for Identity Theft*. Gartner Research.

Likarish, Peter et al. (2008) “B-APT: Bayesian Anti-Phishing Toolbar” In: IEEE International Conference on Communications, Beijing, pp.1745 – 1749.

Chandrasekaran , M.; Narayanan, K. ; Upadhyayas. (2006) “Phishing email detection based on structural properties” In: NYS Cyber Security Conference.

Miyamoto, D.; Hazeyama, H.; Kadobayashi, Y. (2009) “An Evaluation of Machine Learning-Based Methods for Detection of Phishing Sites” In: Proceedings of the 15th International Conference on Advances in Neuro-Information Processing, vol. 1, pp. 539-546.

Ma, Justin et al. (2009) “Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs” In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '09). ACM, New York, pp.1245-1254.

McMillan, R., Gartner: Consumers to lose $2.8 billion to phishers in 2006, *etworkWorld*, 2006.Disponível em: http://www.networkworld.com/news/2006/110906-gartnerconsumers-to-lose-28b.html. Acesso em Setembro 2010.

N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell. Client-side defense against web-based identity theft.” in *NDSS*, 2004. [Online]. Disponível em: http://www.isoc.org/isoc/conferences/ndss/04/proceedings/Papers/Chou.pdf. Acesso em Outubro 2011.

Netcraft toolbar, 2006. [Online]. Disponível em: <http://toolbar.netcraft.com/>

Opendns. PhishTank. Disponível em: <http://www.phishtank.com/>. Acesso em 20 out 2011.

SpoofGuard. Disponível em: <http://crypto.stanford.edu/SpoofGuard/>. Acesso em: 30 outubro 2011.

Phelps, T.A.; R. Wilensky. Robust Hyperlinks and Locations, *D-Lib Magazine*, vol. 6(7/8), 2000.Disponível em: http://www.dlib.org/dlib/july00/wilensky/07wilensky.html

Quinlan, J. R. (1993). “C4.5, Programs for machine learning”. Morgan Kaufmann, San Mateo, Ca.

Richard, Clayton. Insecure real world authentication protocols (or why is phishing so profitable), 2005. Disponível em: http: //www.cl.cam.ac.uk/users/rnc1/phishproto.pdf. Acesso em Dezembro 2011.

Rosiello, Angelo P. E., et al. A Layout-Similarity-Based Approach for Detecting Phishing Pages. In: International Conference ON Security And Privacy Communication Networks, Nice, 3, 2007, p. 454 – 463.

R. Dhamija, J. D. Tygar ; M. Hearst. (2006) ” Why phishing works” In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp. 581–590.

Sheng, Steve, et al. (2009) “An Empirical Analysis of Phishing Blacklists” In: Conference On Email and Anti-Spam,6, Mountain View.

Timko, D. (2008). “The social engineering threat” In: Information Systems Security Association Journal.

Whittaker, Colin; Ryner, Brian; Nazif, Marria. (2010) “Large-Scale Automatic Classification of Phishing Pages” In: Network and Distributed System Security Symposium ,17, San Diego. Proceedings.

Wu, Min.(2006) “Fighting Phishing at the User Interface”. Ph.D. Dissertation. Massachusetts Institute of Technology, Cambridge, MA, USA.

Ye, Cao; Weili, Han; Yueran, Le.(2008) “Anti-phishing Based on Automated Individual White-

List" In: *Proceedings of the 4th ACM workshop on Digital identity management (DIM '08),* ACM, New York, pp.51-60.

Zhang, Yue; Hong, Jason; Cranor, Lorrie. (2007) "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites" In: International Conference On World Wide Web, ACM, New York, n.1357, pp. 639-648.

Zhang, Y., Egelman, S., Cranor, L., Hong, J. Phinding Phish (2007) "Evaluating Anti-Phishing Tools" In: Proceesings of the 14th Annual Network and Distributed System Security Symposium *(NDSS'07).*

Zhang, Jianyi, Luo Shoushan, Gong, Zhe, Ouyang, Xi, Wu, Chaichua, Xin, Yan. (2011)."Protection Against Phishing Attacks: A Survey" …