Lucas Dantas Gama Ayres

Utilizando Aprendizado de Máquina para Detecção Automática de URLs Maliciosas Brasileiras

Salvador - BA

2018

Lucas Dantas Gama Ayres

Utilizando Aprendizado de Máquina para Detecção Automática de URLs Maliciosas Brasileiras

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal da Bahia – Campus Ondina, como requisito parcial para a obtenção do Grau de Bacharel em Ciência da Computação.

Universidade Federal da Bahia

Orientador: Rodrigo Rocha Gomes e Souza

Coorientador: Ítalo Valcy da Silva Brito

Salvador - BA

2018

Lucas Dantas Gama Ayres

Utilizando Aprendizado de Máquina para Detecção Automática de URLs Maliciosas Brasileiras/ Lucas Dantas Gama Ayres. – Salvador - BA, 2018-

63p. : il.

Orientador: Rodrigo Rocha Gomes e Souza

TCC (Graduação) – Universidade Federal da Bahia, 2018.

1. Phishing. 2. Detecção automática. 2. Aprendizado de máquina. I. Rodrigo Rocha Gomes e Souza. II. Universidade Federal da Bahia. III. Faculdade de Ciência da Computação. IV. Utilizando Aprendizado de Máquina para Detecção Automática de URLs Maliciosas Brasileiras

Lucas Dantas Gama Ayres

Utilizando Aprendizado de Máquina para Detecção Automática de URLs Maliciosas Brasileiras

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação da Universidade Federal da Bahia – Campus Ondina, como requisito parcial para a obtenção do Grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Salvador - BA, 02 de agosto de 2018:

Rodrigo Rocha Gomes e Souza
Orientador

Professor
Ivan do Carmo Machado

Professor
Rogerio de Carvalho Bastos

Salvador - BA
2018

Dedico este Trabalho de Conclusão de Curso à minha família e minha noiva, por todo apoio e pela compreensão ao longo do curso e incentivo para ir adiante na realização dos meus sonhos.

Agradecimentos

Agradeço aos meus pais Emanoel e Maria Meire, pois o que estou colhendo hoje são frutos da educação e amor que recebi deles. Vocês são muito especiais para mim e quero poder retribuir tudo que fizeram por mim, amo vocês.

Agradeço também à minha irmã Priscila, por me apoiar e incentivar sempre a fazer as escolhas certas e a continuar persistindo nos meus sonhos.

À Anna Luísa, minha noiva, por todo o amor, carinho, atenção e apoio que me deu em todos os momentos ao longo desses 2 anos juntos. Obrigado por sempre me aconselhar e ajudar a perseguir os meus sonhos, também obrigado por me ajudar com a revisão dos textos. Te amo meu amor!

À Caio Wallison, por me motivar, apoiar e também virar noites na universidade e no trabalho comigo, cada um produzindo sua monografia e incentivando um ao outro.

À equipe do PoP-BA (especialmente Rogerio), por me fornecer apoio e recursos para fazer este projeto viável.

À Madson Araújo, por seu encorajamento e apoio durante o desenvolvimento desta monografia.

Um agradecimento especial ao Professor Dr. Rodrigo Rocha Gomes e Souza por sua inestimável orientação e conhecimentos passados para mim ao longo deste projeto, com toda sua calma e compreensão.

Á Italo Valcy, meu Co-orientador, que me orientou durante todo o trabalho e fez toda a revisão do mesmo.

Aos demais professores, funcionários e alunos da UFBA que, direta ou indiretamente contribuíram de alguma forma. Obrigado a todos.

“Você nunca sabe que resultados virão da sua ação.
Mas se você não fizer nada, não existirão resultados.“
(Mahatma Gandhi)

Resumo

As URLs maliciosas são uma ameaça à segurança cibernética, elas são um canal para atividades criminosas na internet (phishing, spam, malware, etc) e atraem muitos usuários desatentos que se tornam vítimas de golpes, como: roubo de informações privadas e perda de dinheiro. Existem diversos métodos e técnicas de detecção de URLs maliciosas internacionais, mas essas soluções propostas não são eficazes quando se trata das URLs maliciosas que são direcionadas especificamente a comunidade brasileira, pois essas URLs possuem características únicas, contendo marcas de empresas brasileiras nas URLs, o tempo de vida e tamanho da URL é diferente, entre outras características. Foi realizada uma análise para verificar se esses modelos de bases internacionais funcionam bem quando treinados com um conjuntos de URLs nacionais. O resultado obtido mostra uma taxa de acurácia de no máximo 45,26%, comprovando o que foi dito antes, que os métodos existentes não são eficazes, por se tratar de soluções focadas em URLs internacionais. Este trabalho, tem como objetivo criar um método eficaz de detecção de URLs maliciosas, por meio de aprendizado de máquina, focado no cenário nacional. Para alcançar esse objetivo, primeiramente, foi realizada a seleção e extração de 117 características (léxicas, host, blacklist e outras) para obter o máximo de informações das URLs e então empregar 4 classificadores de aprendizado de máquina, incluindo J48, KNN, Naive Bayes e SVM, para poder realizar a comparação de desempenho entre eles e então avaliar a eficácia do modelo. Aplicando-o em conjuntos de dados nacionais reais, composto por 3.950 URLs maliciosas e por 3.162 URLs benignas, foi demonstrado que a abordagem proposta, no cenário nacional, é eficaz na detecção de URLs de phishing, com uma taxa precisão e acurácia acima de 96%, demonstrando ser melhor do que muitos trabalhos já existentes na literatura.

Palavras-chave: URL, Aprendizado de máquina, Phishing, Características, Classificadores.

Abstract

Malicious URLs are a cyber security threat, they are a conduit for criminal activity on the internet (phishing, spam, malware, etc.) and attract many inattentive users who become victims of scams, such as: theft of private information and loss of money. There are several methods and techniques for detecting malicious URLs internationally, but these proposed solutions are not effective when it comes to malicious URLs that are targeted specifically to the Brazilian community, since these URLs have unique characteristics, containing Brazilian company trademarks in URLs, lifetime and URL size is different, among other features. An analysis was performed to see if these international base models work well when trained with a set of national URLs. The result obtained shows an accuracy rate of at most 45.26%, confirming what has been said before, that the existing methods are not effective because they are solutions focused on international URLs. This work aims to create an efficient method of detecting malicious URLs, through machine learning, focused on the national scenario. In order to reach this goal, the selection and extraction of 117 characteristics (lexical, host, blacklist and others) was carried out to obtain the maximum information of the URLs and then to use 4 machine learning classifiers, including J48, KNN, Naive Bayes and SVM, in order to be able to compare the performance between them and then evaluate the effectiveness of the model. Applying it to real national data sets, consisting of 3,950 malicious URLs and 3,162 benign URLs, it has been demonstrated that the proposed approach in the national scenario is effective in detecting phishing URLs with a precision and accuracy rate of over 96%, proving to be better than many existing works in the literature.

Keywords: URL, Machine Learning, Phishing, Features, Classifiers.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ASN	Autonomous System Number
CAIS	Centro de Atendimento a Incidentes de Segurança da RNP
CaUMa	Catalogo de URLs Maliciosas
CNAME	Canonical Name
CSV	Comma-Separated Values
DMOZ	Directory Mozilla
DNS	Domain Name System
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ICANN	Internet Corporation for Assigned Names and Numbers
KNN	k-nearest neighbors
MX	Mai Exchange
NS	Name Server
PoP-BA	Ponto de Presenca da RNP na Bahia
PTR	Pointer
RNP	Rede Nacional de Ensino e Pesquisa
RR	Registro de recurso
SEO	Search Engine Optimization
SMS	Short Message Service
SPF	Sender Policy Framework
SRM	Structural Risk Minimization
SVM	Support Vector Machine

TLD	Top Level Domain
TXT	Text
UFBA	Universidade Federal da Bahia
URL	Uniform Resource Locator
Web	World Wide Web
WoT	Web of Trust

Sumário

1 Introdução

Phishing é uma técnica que usa engenharia social para fazer vítimas, enganando-as com o objetivo de obter suas informações pessoais (geralmente de cunho financeiro) e depois causar-lhes prejuízos (OLIVO, 2010). Na Internet, o *phishing* pode chegar ao usuário (vítima) de várias maneiras, como janela *pop-up* no navegador (*browser*), mensagens instantâneas, e-mails e redes sociais (OLIVO, 2010). Atualmente muitos usuários de Internet são alvos de ataques de *phishing* e o Brasil está no topo da lista desde 2015 (Kaspersky Lab, 2017a).

Geralmente, a vítima é convencida a clicar em um link, que descarregará e instalará algum *malware* (código malicioso) ou apresentará uma página falsa que imita a imagem de uma empresa famosa e confiável para poder chamar a atenção das vítimas e levá-las a divulgar informações confidenciais, como senhas de banco e dados de cartão de crédito. Os *malwares* são programas que têm como objetivo infectar, danificar e capturar informações de um computador de forma ilícita (ALECRIM, 2017).

De acordo com um estudo feito pela Kaspersky (Kaspersky Lab, 2017b), durante o terceiro trimestre de 2017 foram registrados mais de 59 milhões de detecções pelo sistema *antiphishing*. O Brasil continua a ser líder no ranking de usuários atacados por esses tipos de ameaças: os brasileiros representam cerca de 19,95% dos ataques no índice mundial. Essas fraudes direcionadas à comunidade brasileira estão evoluindo dia após dia, se tornando cada vez mais sofisticadas e isso faz com que os usuários tornem-se mais suscetíveis a esses tipos de golpes na Internet.

Outro estudo, feito também pela Kaspersky, mostra que, em 2016, 77,26% dos ataques foram a partir de URLs, que equivalem a um total de 261.774.932 URLs únicas, o restante dos ataques foi a partir de malwares (Kaspersky Lab, 2016). Esse resultado mostra que a URL é o meio mais utilizado para se conseguir atacar um usuário de Internet e por isso a URL deve ser estudada mais a fundo, suas características, formato e aspectos para tentar chegar a um denominador comum que ajude a automatizar a detecção dessas fraudes.

Sistemas eficazes para detecção dessas URLs maliciosas em tempo hábil podem ajudar muito no combate a essas ameaças à segurança cibernética. Consequentemente, pesquisadores e profissionais trabalharam para projetar soluções efetivas para detecção de URLs maliciosas. Várias abordagens tem sido utilizadas para enfrentar o problema de Detecção de URLs maliciosas. De acordo com os princípios fundamentais, essas abordagens podem ser agrupadas em duas categorias: (i) *blacklist* (lista negra), e (ii) abordagens de aprendizado de máquina (Canali et al, 2011), (Eshetet et al, 2013).

O método mais comum para detectar URLs maliciosas implantados por muitos grupos de antivírus é o método da *blacklist*. *Blacklists* são sistemas que mantêm listas de URLs que foram

analisadas previamente e classificadas como danosas aos usuários. Essas listas são mantidas em um banco de dados. Quando se utiliza um serviço de *blacklist*, cada site visitado é verificado com esse banco de dados e então o acesso é permitido ou negado com base na classificação de segurança do site. Tal técnica é extremamente rápida devido a uma simples sobrecarga de consulta e, portanto, é muito fácil de implementar. No entanto, é quase impossível manter uma lista exaustiva de URLs maliciosas, especialmente porque novas URLs são geradas todos os dias. Os atacantes usam técnicas criativas para evadir as listas negras e enganar os usuários, modificando a URL para "parecer" legítima através de ofuscação (Garera et al, 2007).

Para superar essas questões, na última década, os pesquisadores aplicaram técnicas de aprendizado de máquina para detecção de URLs maliciosas (PATIL; PATIL, 2015), (Garera et al, 2007), (MCGRATH; GUPTA, 2008). Onde utiliza-se um conjunto de URLs como dados de treinamento e, com base nas propriedades estatísticas, aprenda uma função de predição para classificar uma URL como maliciosa ou benigna. Isso lhes dá a capacidade de generalizar para novas URLs ao contrário dos métodos de *blacklist*.

Em 2015 foi realizado um estudo pelos membros do PoP-BA/RNP (*Ponto de Presença da RNP na Bahia*) e da UFBA (*Universidade Federal da Bahia*) onde alguns serviços de *blacklist* de URLs foram testados e analisados com cerca de 1920 URLs maliciosas direcionadas à comunidade brasileira, coletadas no período de um ano e armazenadas no CaUMa (*Catálogo de URLs Maliciosas*), que é um serviço desenvolvido e mantido pelo PoP-BA, RNP e UFBA onde são catalogadas URLs maliciosas direcionadas à comunidade brasileira. Analisando os resultados, foi identificado que a porcentagem de detecção foi insatisfatória: menos de 9% de um total de 1920 URLs foram detectadas como fraude. Com isso foi observado que esses serviços são pouco eficazes quando se trata de URLs que são direcionadas para a comunidade brasileira (BRITO et al., 2015).

Outro estudo também realizado pelos membros do PoP-BA/RNP e da UFBA em 2016, mostra que as soluções existentes não possuem bons resultados de detecção pois as URLs maliciosas que são direcionadas à comunidade brasileira possuem características únicas, como: nomes de marcas de empresas brasileiras contidos na URL, a maioria das URLs possuem entre 50 e 100 caracteres, o tempo de vida das URLs de phishing é igual ou superior a 5 dias, sendo que na literatura aponta que cerca de 70% das URLs de phishing permanecem menos de 48 horas online. Isso mostra que realmente é necessário a criação de um método de detecção focado nas URLs direcionadas a comunidade brasileira (BRITO et al., 2016).

Diante deste problema, foi apresentado uma metodologia heurística para classificar automaticamente URLs como sendo phishing ou benigna. Essa metodologia pode ser utilizada para impedir ataques de phishing ou alertar os usuário sobre uma potencial ameaça. Como o foco é na própria URL, esse modelo pode ser aplicado em qualquer lugar que uma URL possa ser incorporada, como no e-mail, páginas da web, chat, etc. O modelo foi avaliado em conjuntos de dados nacionais reais com 3.950 URLs de phishing e 3.162 URLs benignas. Também foi

demonstrado experimentalmente que o modelo apresenta uma alta taxa de precisão e acurácia.

No que diz respeito a contribuições, este trabalho apresenta: (i) um conjunto de características utilizadas na detecção de URLs maliciosas; (ii) um software que possibilita a extração de características de URLs; (iii) Conjunto de URL de bases nacionais e internacionais; (iv) Dataset; (v) Análise das características; (vi) Análise de modelos treinados com bases internacionais em dados nacionais; (vii) Análise de desempenho dos classificadores; (viii) Árvore de decisão.

1.1 Objetivo Geral

Este trabalho tem como objetivo criar um método de detecção automática de URLs maliciosas que são direcionadas ao público brasileiro, por meio de estudos sobre as características dessas URLs e comparação de diferentes algoritmos de aprendizado de máquina.

1.2 Objetivos Específicos

- Construir banco de dados contendo URLs maliciosas e URLs benignas direcionadas à comunidade brasileira;
- Estudar e selecionar quais são as características relevantes que podem ajudar a identificar uma URL como maliciosa;
- Desenvolver um software capaz de realizar a extração automática das características dessas URLs;
- Avaliar diferentes algoritmos e compará-los para identificar o que oferece melhores resultados;
- Avaliar se modelos treinados com bases internacionais funcionam bem nos dados nacionais;

1.3 Organização do trabalho

Este trabalho está estruturado como segue: no Capítulo 2 são apresentados os conceitos de *phishing*, *malware*, *blacklist*, aprendizado de máquina e demais conteúdos importantes para o entendimento do trabalho. No Capítulo 3, discute os trabalhos relacionados. No Capítulo 4, será mostrada uma descrição detalhada dos métodos utilizados nesse trabalho, bem como do processo de desenvolvimento do software de extração de características das URLs, construção do dataset, experimentos e mais. No Capítulo 5 exibirá os resultados obtidos e suas respectivas discussões. A conclusão e direcionamento para trabalhos futuros estão no Capítulo 6.

2 Fundamentação Teórica

Neste capítulo são apresentados os principais fundamentos teóricos envolvidos no processo de detecção de URLs maliciosas utilizando técnicas de aprendizado de máquina.

2.1 URL

URL (*Uniform Resource Locator*), que em português significa Localizador Padrão de Recursos, é o endereço de um recurso, seja ele uma impressora, um computador ou um site na Internet. Tecnicamente um URL é uma *string* compacta (BERNERS-LEE; MASINTER; MCCAHILL, 1994). Geralmente uma URL faz referência a páginas WEB (HTTP), mas também pode fazer referência a transferência de arquivos (FTP), e-mail (mailto) e muitas outras aplicações. Nessa monografia o foco será em endereços de sites na Internet (http).

2.1.1 Estrutura das URLs

Uma URL possui a seguinte estrutura (BERNERS-LEE; FIELDING; MASINTER, 2005):

Figura 1: Estrutura de uma URL

1. **Esquema:** É o responsável por especificar o protocolo que será utilizado para acessar o recurso que faz referência no resto da URL. O padrão dentro de uma URL para um esquema é iniciando sempre com uma letra e separado com dois pontos (“:”). Nessa monografia, o protocolo pode ser HTTP ou HTTPS (Protocolos que permitem *web servers* e *browsers* enviarem e receberem dados através da Internet). Existem diversos outros protocolos, porém que estão fora do escopo deste trabalho. protocolos, tais como: (i) FTP, que é utilizado para transferência de arquivos; e (ii) TELNET que trata-se de um protocolo de uso geral cujo objetivo é permitir a conexão, via terminal, de um cliente remoto em um dado servidor, entre outros.

2. **Autoridade:** A autoridade de uma URL é composta pelas informações do usuário (opcional), host e a porta (opcional). Abaixo é possível saber mais detalhes sobre cada um desses componentes:

a) **Informações do usuário (usuário e senha):** O usuário e senha são componentes opcionais da parte de autoridade da URL. O usuário e senha é requerido quando se faz necessário alguma autenticação para acesso ao recurso, e sua notação é simples, tendo sempre o usuário e senha separado por dois pontos (“:”), e a utilização de um arroba (“@”) para indicar o início do host. O uso do formato “usuário:senha” na parte de autoridade da URL, está obsoleto, pois a passagem de informações de autenticação em texto claro é um risco de segurança em quase todos os casos em que é usado.

b) **Host:** O host pode ser identificado por um nome (ex: www.google.com.br) ou um endereço IP (ex: 172.217.30.3) do servidor da web a ser acessado, o nome do host não faz distinção entre letras maiúsculas e minúsculas. Esse nome consiste em uma sequência de rótulos de domínio separados por “.”, Cada rótulo de domínio começando e terminando com um caractere alfanumérico. O nome do host é composto por:

   i. **Domínio:** O domínio é o nome de identificação de um site na internet, por exemplo, startonapp.com.br. Ele é formado pelo nome e pela extensão (O nome técnico desta extensão é TLD ou *Top Level Domain*): startonapp é o nome do domínio e o .com.br é a extensão. Tecnicamente, o domínio é chamado de *second level domain* (domínio de segundo nível), ele é único e para ser usado é necessário registrá-lo.

   ii. **Subdomínio:** É um endereço que faz parte do domínio, ou seja, é uma ramificação do domínio. Geralmente é possível criar quantos subdomínios quiser a partir de um domínio. O subdomínio fica à esquerda do domínio, antes do primeiro ponto, por exemplo, blog.startonapp.com.br, onde blog é o subdomínio.

   iii. **Top Level Domain (TLD):** Que em português significa Domínio de Topo, é o sufixo de um endereço na web. O TLD é a última parte do nome de domínio. Um TLD identifica algo sobre o site associado a ele, como seu objetivo, a organização que possui ou a área geográfica onde se origina. Cada TLD possui um registro separado gerenciado por uma organização designada sob a direção da *Internet Corporation for Assigned Names and Numbers* (ICANN).

c) **Porta:** A porta é o ponto lógico no qual se pode executar a conexão com o servidor. Quando a porta não está presente, uma porta padrão para o esquema específico é assumida. Por exemplo, porta 80 para http ou 443 para https. (Opcional)

O componente de autoridade é precedido por uma barra dupla (“//”) e é finalizado pela próxima barra (“/”), ponto de interrogação (“?”), caractere de sinal numérico (“#”) ou pelo final da URL.

3. **Caminho:** O caminho consiste em uma sequência de segmentos de caminho separados por um caractere de barra (“/”). Um caminho é sempre definido em uma URL, embora o

caminho definido possa estar vazio (comprimento zero). O caminho identifica um recurso específico no host que o cliente web deseja acessar.

4. **Query:** Query ou Parâmetros de consulta, contém dados não-hierárquicos que, juntamente com dados do caminho, serve para identificar um recurso dentro do escopo do esquema da URL e autoridade (caso existam). O parâmetro de consulta é opcional e é indicado pelo primeiro caractere (“?”) e terminado por um sinal (“#”) ou até o final da URL. É utilizado para especificar algo para a aplicação que está prestes a ser requisitada.

5. **Fragmento:** Um fragmento é opcional, é muito comum utilizar este componente quando é necessário navegar por trechos específicos em uma página HTML. Com este recurso é possível criar componentes como âncoras na página, e então aliado ao símbolo (“#”) é possível ao carregar o HTML navegar direto para um determinado trecho.

O esquema informa ao computador como conectar-se, o domínio especifica onde conectar-se e os demais componentes da URL especificam o que está sendo solicitado.

2.2 Phishing

Phishing é o tipo de fraude por meio da qual um golpista tenta obter dados pessoais e financeiros de um usuário, pela utilização combinada de meios técnicos e engenharia social (CGI.BR, 2012). Os meios mais utilizados nesse tipo de fraude são a aparente comunicação oficial, geralmente realizada por e-mail, bem como as páginas falsas ou clonadas da Internet onde a vítima, sem ter a consciência do perigo, acaba passando dados sensíveis e confidencias, ou até mesmo quando o internauta, seja por curiosidade, por caridade ou pela possibilidade de obter alguma vantagem financeira acaba sendo vítima de golpe eletrônico (CGI.BR, 2012).

Outros ataques que não são muito conhecidos, mas que também acontecem, são os voltados para o sistema de SMS, conhecido como *smishing*, ou aqueles feitos por meio de telefonema, batizado, por sua vez, *devishing* (DIEZ, 2013).

Uma pesquisa realizada em 2016 pela Kaspersky Lab, que é uma das maiores empresas de segurança de dados do mundo, mostrou quais foram os países que sofreram mais ataques de *phishing*. Esses dados ajudam a entender a dimensão do *phishing* nos dias atuais. Através de monitoramento de seus softwares, a companhia descobriu que, somente no primeiro trimestre de 2016, seu sistema *anti-phishing* foi disparado cerca de 35 milhões de vezes e o país mais atingido foi o Brasil (AO Kaspersky Lab, 2016). Esses dados estão ilustrados na Figura 2 e na Tabela 1.

Através desses dados é possível notar que o *phishing* é um ataque muito difundido e presente na vida dos usuários de Internet, especialmente no Brasil que lidera esse ranking. Isso implica na necessidade de compreender cada vez mais o que está por trás desse tipo de fraude.

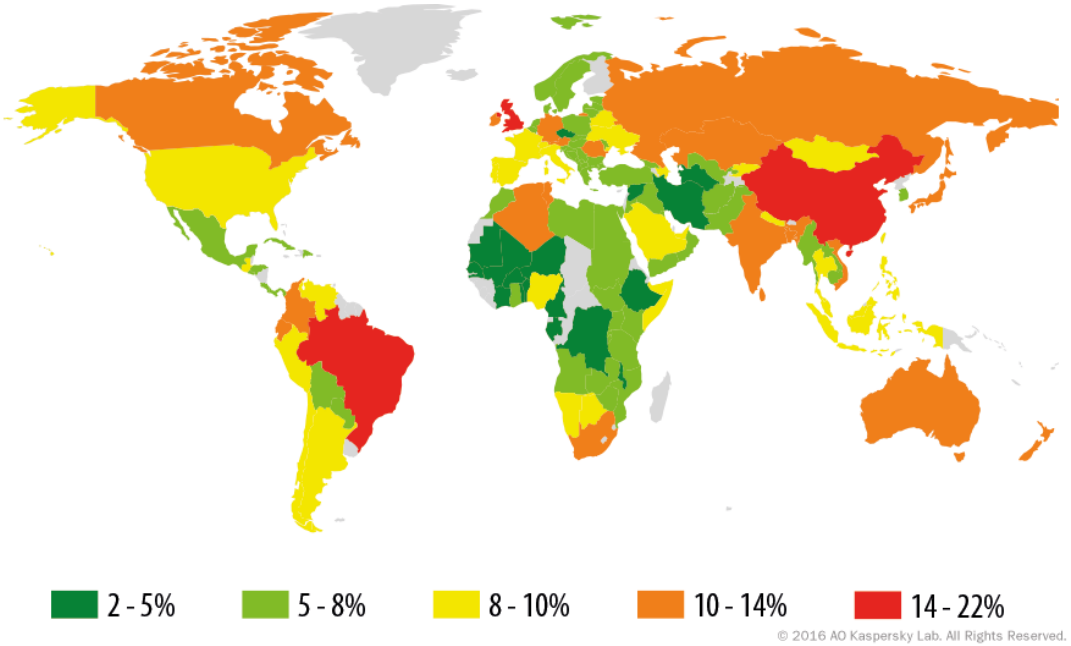

Figura 2: Geografia de ataques *phishing* no primeiro trimestre de 2016. (AO Kaspersky Lab, 2016)

Tabela 1: Top 10 países mais afetados por ataques

Brasil	**21.5%**
China	16.7%
Reino Unido	14.6%
Japão	13.8%
Índia	13.1%
Austrália	12.9%
Bangladesh	12.4%
Canadá	12.4%
Equador	12.2%
Irlanda	12.0%

O e-mail é bastante utilizado para disseminação de fraudes eletrônicas (OLLMANN, 2007), sabendo disso, o PoP-BA/RNP (*Ponto de Presença da RNP na Bahia*) em parceria com o CAIS/RNP (*Centro de Atendimento a Incidentes de Segurança da RNP*), que realizam um trabalho de tratamento e análise de fraudes enviadas por e-mail, fizeram um estudo (BRITO et al., 2015) onde foi constatado que cerca de 90% dos e-mails analisados por eles, continham URLs para sites ou arquivos maliciosos no corpo da mensagem, mostrando que os atacantes estão enviando cada vez mais, mensagens contendo URLs maliciosas. Diante dessa informação, é possível ver que ferramentas de análise de URLs é um importante mecanismo de proteção para os usuários.

2.3 Malware

Códigos maliciosos (*malware*) são programas especificamente desenvolvidos para executar ações danosas e atividades maliciosas em um computador (CGI.BR, 2012). Algumas das diversas formas como os códigos maliciosos podem infectar ou comprometer um computador são:

- Pela exploração de vulnerabilidades existentes nos programas instalados;
- Pela auto-execução de mídias removíveis infectadas, como pen-drives;
- Pelo acesso a páginas Web maliciosas, utilizando navegadores vulneráveis;
- Pela ação direta de atacantes que, após invadirem o computador, incluem arquivos contendo códigos maliciosos;
- Pela execução de arquivos previamente infectados, obtidos em anexos de mensagens eletrônicas, via mídias removíveis ou diretamente de outros computadores (através do compartilhamento de recursos);
- Pela ação do usuário ao clicar em links desconhecidos que os leva ao download e execução de código malicioso;

Além da questão das novas formas de ataques, o número crescente desses códigos maliciosos representa uma preocupação, não somente para as empresas, mas também para entidades governamentais, que devem se preocupar com ameaças que podem causar danos em sistemas críticos de um país e afetar diretamente a segurança física do cidadão desse país (Finjan Research Center, 2009).

(BRITO et al., 2015) mostram que os atacantes estão enviando cada vez mais mensagens com links para baixar um código malicioso, ao invés de colocar o código malicioso diretamente no corpo do e-mail e com isso os atacantes podem fazer evasão de filtros.

2.4 DNS - *Domain Name System*

O protocolo DNS (Domain Name System) é um sistema de nomes de domínio que realiza o mapeamento de endereços IP para nomes de domínio e vice-versa. Através deste sistema de nomes de domínio é possível acessar sites sem haver a necessidade de inserir o seu endereço IP, e ao invés disso basta apenas saber o nome do site (DANIEL, 2006).

O DNS armazena as informações na forma de zonas. Em uma zona (O termo zona se refere a informações contidas em um arquivo do banco de dados do DNS) pode haver informações sobre um ou mais domínios. As informações são adicionadas em uma zona do DNS, através

da criação de registros de recursos (RR). Cada domínio tem uma série de registros de recursos (RR) como endereços de IP, serviço de e-mails e a própria resolução de nomes. Os registros de recursos são responsáveis por informar qual tipo de mapeamento será feito para um determinado domínio. Os registros de recurso são armazenados no formato binário internamente para o uso do software de DNS. Abaixo é possível ver de forma detalhada sobre alguns dos tipos mais importantes de registros de recurso (DANIEL, 2006).

- Registro A:

  O registro do tipo A é muito utilizado em arquivos de zona, pois ele é responsável por realizar o mapeamento entre um nome de domínio e um endereço IP versão 4 (IPv4).

- Registro AAAA:

  O registro do tipo AAAA executa a mesma função do registro do tipo A, porém, para um endereço de IV versão 6 (IPv6).

- Registro NS (*Name Server*):

  O registro do tipo NS é utilizado para especificar os servidores DNS que tem autoridade sobre determinada zona. Pelo menos, dois registros do tipo NS devem ser definidos para cada domínio. Geralmente, um principal e outro secundário.

- Registro CNAME (*Canonical Name*):

  O registro do tipo CNAME especifica um apelido (alias) ou nomes alternativos para um host. É uma forma de redirecionamento.

- Registro MX (*Mai Exchange*):

  O registro do tipo MX aponta o servidor de e-mails. Pode-se especificar mais de um endereço, formando-se assim uma lista em ordem de prioridade para que haja alternativas no caso de algum e-mail não puder ser entregue.

- Registro PTR (*Pointer*):

  Os registros do tipo PTR são utilizados para informar quais são os endereços reversos de uma zona reversa. Esse registro será utilizado para realizar o mapeamento de endereços IP em nomes DNS.

- Registro TXT (*Text*):

  O registro do tipo TXT permite associar um texto curto com um *hostname*. É possível ter múltiplos registros do tipo TXT para um nome de host único.

- Registro SPF (*Sender Policy Framework*):

  O registro do tipo SPF (*Sender Policy Framework*) identifica quais servidores de e-mail têm permissão para enviar e-mails em nome de um domínio.

(GöRLING, 2007) SPF é uma abreviação de “Sender Policy Framework”, que é um sistema que evita que outros servidores enviem e-mails não autorizados em nome de outro domínio. Um registro SPF é publicado pelo proprietário do domínio, informando quais servidores de e-mail (endereços IP) estão autorizados a enviar um e-mail para o nome de domínio específico. O servidor de e-mail do destinatário verifica o endereço do remetente com as informações publicadas pelo proprietário do nome de domínio para garantir que o e-mail esteja vindo dos servidores autorizados. Somente quando os detalhes coincidirem, o e-mail é entregue. Se o nome de domínio tiver um registro SPF válido, uma entidade não autorizada não poderá enviar e-mails em nome desse domínio.

(GöRLING, 2007) O SPF é um mecanismo especificamente projetado para mitigar problemas de fraude e phishing. É usado para validar que a mensagem foi enviada pelo domínio do remetente especificado no endereço “MAIL FROM:” do e-mail. Por exemplo, quando um e-mail de paypal.com solicita a revalidação de usuário e senha, o SPF pode verificar se foi realmente enviado a partir do domínio paypal.com ou é parte de um golpe.

2.5 WHOIS

(DAIGLE, 2004) O WHOIS é uma base de dados distribuída e provida por entidades de registro que permite obter informações de recursos da internet, tais como: Domínios de redes, endereçamentos IPs e Sistemas Autônomos (AS).

Não existe um padrão para as informações que são disponibilizadas pelos serviços de WHOIS, existe um subconjunto de informações típicas que podem ser encontradas na maiorias das bases de dados WHOIS, como:

- Contato técnico
- Servido DNS autoritário
- Bloco de endereçamento IP
- Informações sobre o detentor do bloco
- Servidores de nome (NS)
- Data de criação do domínio
- Data da última renovação do domínio
- Data de expiração do domínio
- Informações sobre um WHOIS adicional

Pode-se utilizar diferentes interfaces para realizar uma consulta na base de dados WHOIS, entre elas, interface web ou por meio de aplicativos nos sistemas operacionais.

2.6 ASN - *Autonomous System Number*

(HAWKINSON; BATES, 1996) Um ASN (*Autonomous System Number*) ou AS (*Autonomous System*), é um grupo conectado de um ou mais prefixos IP, executados por um ou mais operadores de rede que possuem uma política de roteamento simples e bem definida, ele identifica exclusivamente a organização responsável por um bloco de endereços IP. Um ASN tem um número globalmente exclusivo associado a ele, este número é usado na troca de informações de roteamento externas (entre ASes vizinhos) e como um identificador do próprio AS.

2.7 Princípios de detecção de URLs maliciosas

Várias abordagens tem sido utilizadas para enfrentar o problema de detecção de URLs maliciosas, essas abordagens podem ser agrupadas em duas categorias: (i) *blacklist* (lista negra) ou heurística, e (ii) abordagens de aprendizado de máquina (Canali et al, 2011), (Eshetet et al, 2013). Essas categorias são descritas abaixo:

- ***Blacklist* (lista negra) ou abordagens heurísticas:**

  Abordagens que utilizam *blacklists* são uma técnica bastante comum e clássica para a detecção de URLs maliciosas, que mantêm uma lista de URLs que são conhecidas como maliciosas.

  Sempre que uma nova URL for visitada, é realizada uma consulta de banco de dados. Se a URL for encontrada na *blacklist*, então ela é considerada maliciosa e, em seguida, um alerta é exibido; senão é assumida como benigna. É difícil manter uma *blacklist* de todas as possíveis URLs maliciosas, devido ao rápido aumento de URLs que são geradas diariamente, tornando impossível a detecção de novas ameaças (Sheng et al, 2009). Isto é bastante preocupante quando os atacantes geram novas URLs algoritmicamente, e podem então ignorar todas as *blacklists*. Apesar de vários problemas enfrentados pela *blacklist* (Sinha et al, 2008), devido à sua simplicidade e eficiência, elas continuam sendo uma das técnicas mais utilizadas, inclusive são utilizadas por vários sistemas de anti-vírus.

  As abordagens heurísticas (Seifert et al, 2008) são um tipo de extensões de métodos baseados em *blacklists*, onde a ideia é criar uma "*blacklist* de assinaturas". São identificados os ataques comuns e, com base em seus comportamentos, uma assinatura é atribuída a esse tipo de ataque. Os Sistemas de Detecção de Intrusão podem escanear as páginas da Web para tais assinaturas e se algum comportamento suspeito for encontrado é criado um código (flag) que representa esse comportamento. Esses métodos possuem melhores recursos de generalização do que a *blacklist*, pois eles também têm a capacidade de detectar ameaças em novas URLs. No entanto, esses métodos podem ser projetados apenas para um número limitado de ameaças comuns e não podem se generalizar para todos os tipos de ataques (novos) (Seifert et al, 2008).

- **Abordagens de Aprendizado de Máquina:**

  Essas abordagens analisam as informações de uma determinada URL e suas páginas web correspondentes, extraindo características, treinando um modelo e fazendo previsões em um conjunto de dados de URLs maliciosas e benignas. Existem dois tipos de características que podem ser utilizados: características estáticas e características dinâmicas. Na análise estática, é realizado a análise de uma página web com base em informações disponíveis sem precisar executar a URL (ou seja, executando JavaScript ou outro código) (Ma et al, 2009a). As características extraídas incluem recursos lexicais da cadeia (string) da URL e informações sobre o host. Como nenhuma execução é necessária, esses métodos são mais seguros do que as abordagens dinâmicas. A distribuição dessas características é diferente para URLs maliciosas e benignas. Usando esta informação de distribuição, um modelo de predição pode ser construído, o que pode fazer previsões em novas URLs. As técnicas de análise dinâmica incluem o monitoramento do comportamento das URLs, procurando qualquer anomalia, como por exemplo, múltiplos redirecionamentos, longa duração de carregamento, download de um arquivo, etc. As técnicas de análise dinâmica têm riscos inerentes e são difíceis de implementar e generalizar.

  Uma visão geral da estrutura de detecção de URLs maliciosas utilizando aprendizado de máquina pode ser visto na Figura 3. Primeiro, é realizada a coleta de URLs maliciosas e URLs benignas. Em seguida, é necessário extrair uma série de características, para classificar as instâncias em suas classes correspondentes (URLs de phishing pertencentes à classe positiva e URLs benignas pertencentes à classe negativa). Em seguida, são aplicados vários algoritmos de aprendizado de máquina para criar modelos a partir de dados de treinamento, que são compostos de pares de valores de características e rótulos de classes. Um conjunto separado de dados de teste é então fornecido aos modelos e a classe prevista da instância de dados é comparada à classe real dos dados para calcular a precisão dos modelos de classificação.

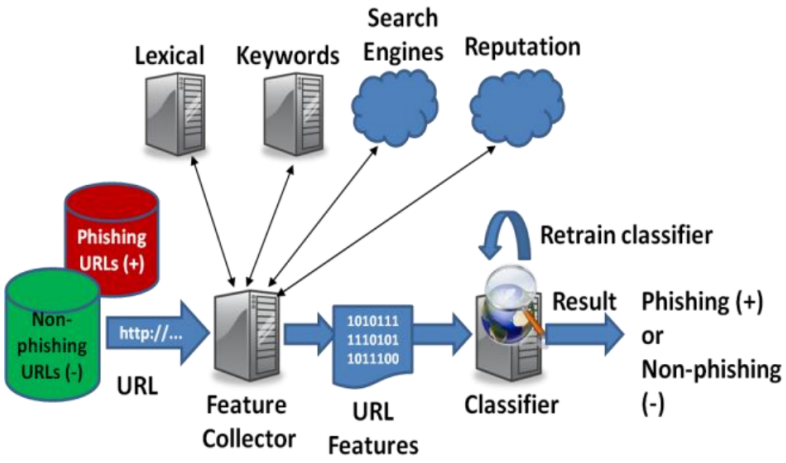

Figura 3: Visão geral da estrutura de detecção de URL de phishing. (Basnet et al, 2014)

2.8 Representação de Características

O sucesso de um modelo de aprendizado de máquina depende muito da qualidade do conjunto de dados de treinamento, que, por sua vez, depende da qualidade da representação de características. Dado uma URL u $\in \mathbb{U}$, onde $\mathbb{U}$ denota um domínio de quaisquer cadeias de URLs válidas, o objetivo da representação de característica é encontrar um mapeamento g: $\mathbb{U} \rightarrow \mathbb{R}^d$, tal que g(u) $\rightarrow$ X onde X $\in \mathbb{R}^d$ é um vetor de características d-dimensional, que pode ser alimentado em modelos de aprendizado de máquinas. O processo de representação de características pode ser dividido em duas etapas:

- **Extração das características:**

  Esta fase visa coletar a maioria, se não todas, as informações relevantes sobre a URL. Isso inclui informações como a quantidade de redirecionamentos de uma URL, as características diretas da URL, como a existência de palavras-chave na URL, quantidade de caracteres especiais na URL e as informações sobre o host, o conteúdo do site, como HTML e JavaScript, informações de popularidade, etc.

- **Pré-processamento das características:**

  Nessa fase, as informações não estruturadas sobre a URL (por exemplo, descrição textual) são adequadamente formatadas e convertidas em um vetor numérico para que possam ser alimentadas em algoritmos de aprendizado de máquina. Além disso, algumas normalizações de dados geralmente podem ser usadas para lidar com o problema de escala.

Para a detecção de URLs maliciosas, neste trabalho foram propostos vários tipos de características que podem ser usados para fornecer informações relevantes acerca da reputação da URL. Nesta monografia, essas características foram categorizadas em: características baseadas em *blacklist*, características léxicas baseadas na URL, características baseadas em host e outros (Indexação no google, redirecionamentos, etc.). Em seguida, discutiremos cada uma dessas categorias de características com mais detalhes.

1. **Características baseadas em *blacklists*:**

   Como mencionado anteriormente, uma técnica trivial e bastante utilizada para identificar URLs maliciosas é o uso de backlists. Uma URL que é identificada como maliciosa (seja através de extensas análises ou avaliações de pessoas) entra na lista. No entanto, foi observado que a *blacklist*, apesar de sua simplicidade e facilidade de implementação, sofre de falso-negativos não triviais (Sinha et al, 2008), devido à dificuldade em manter listas atualizadas exaustivas. Consequentemente, ao invés de usar a presença da *blacklist* sozinha como um tomador de decisão, ela pode ser usada como uma característica poderosa. Em particular, foi usado a presença em uma *blacklist* como característica, de 6 provedores

diferentes de serviços de *blacklist*. Eles também analisaram a eficácia desses recursos em comparação com outros recursos e observaram que os recursos da *blacklist* por si só não tinham um desempenho tão bom quanto outros recursos, mas quando usados em conjunto com outros recursos, o desempenho geral do modelo de previsão melhorou (Ma et al, 2009a).

(Prakash et al, 2010) Para evitar a detecção através de uma *blacklist*, muitos atacantes fizeram pequenas modificações na URL original. Eles propuseram estender a *blacklist* ao derivar novas URLs com base em cinco heurísticas, incluindo: Substituição de domínios de nível superior (TLDs), equivalência de endereço IP, semelhança da estrutura de diretórios, substituição de cadeia de caracteres e equivalência de marcas (Exemplo: BB, ricardoeletro, magazineluiza, etc), fazendo com que uma URL maliciosa seja despercebida.

2. **Características Léxicas:**

   As características léxicas são características obtidas com base nas propriedades do nome da URL, onde a cadeia da URL deve ser processada para extrair características úteis. A motivação é que, com base em como a URL "parece", deve ser possível identificar a natureza maliciosa de uma URL. Por exemplo, muitos métodos de ofuscação tentam "parecer" como URLs benignas imitando seus nomes e adicionando uma pequena variação a ele. Na prática, essas características léxicas são usadas em conjunto com várias outras características (por exemplo, características baseadas em host) para melhorar o desempenho do modelo. Em seguida, analisamos algumas das características léxicas utilizadas para a detecção de URLs maliciosas.

   Características Léxicas Tradicionais: as características léxicas mais utilizadas incluem propriedades estatísticas da cadeia da URL, como o comprimento da URL, o comprimento de cada um dos componentes da URL (Hostname, Top Level Domain, Primary domain, etc), o número de caracteres especiais, etc. (Kolari et al, 2006) Para extrair palavras da string de uma URL, a sequência de caracteres foi processada de tal forma que cada segmento delimitado por um caractere especial (por exemplo, "/", ".", "?", "=", Etc.) compreendeu uma palavra. Com base em todos os diferentes tipos de palavras em todas as URLs, um dicionário foi construído, ou seja, cada palavra se tornou uma característica. Se a palavra estivesse presente na URL, o valor da característica seria 1 e 0 caso contrário. Isso também é conhecido como modelo de bag-of-words.

3. **Características baseadas em Host:**

   As características baseadas no host são obtidas das propriedades do nome do host da URL (Ma et al, 2009b). Elas nos permitem conhecer a localização dos hosts maliciosos, a identidade, o estilo de gerenciamento e as propriedades desses hosts.

   Foi realizado um estudo sobre o impacto de algumas características baseadas no host sobre as URLs maliciosas. Algumas das principais observações foram que os phishers

exploravam os serviços de encurtadores de URLs; o tempo de vida do registro do domínio foi quase imediato para as URLs maliciosas; e muitos usaram botnets para hospedar-se em várias máquinas de vários países. Consequentemente, os recursos baseados no host se tornaram um elemento importante na detecção de URLs maliciosas (MCGRATH; GUPTA, 2008).

(MCGRATH; GUPTA, 2008; Ma et al, 2009a; Ma et al, 2009b) Alguns pesquisadores propuseram o uso de várias características baseadas no host, incluindo: propriedades de endereço IP, informações do WHOIS, localização, propriedades de nomes de domínio e velocidade de conexão. As propriedades do endereço IP compreendem as características obtidas a partir do prefixo do endereço IP e do número do sistema autônomo (AS). Isso inclui se os IPs dos registros A, MX ou NS estão nos mesmos ASes ou prefixos um como o outro. As informações de WHOIS incluem datas de registro de nomes de domínio, registradores e registrantes. A informação de localização compreende a localização geográfica (país/cidade) a que o endereço de IP pertence. As propriedades do Nome de Domínio incluem valores de tempo de vida, presença de determinadas palavras-chave como "cliente" e "servidor", se o endereço IP estiver no nome do host ou não e o registro PTR resolva um dos endereços IP do host. Uma vez que muitas das características são informações relacionadas à identidade, é necessário uma abordagem de bag-of-words para armazená-las em um vetor numérico, onde cada palavra corresponde a uma identidade específica. Como as características léxicas, a adoção de tal abordagem leva a uma grande quantidade de características. Para as 2 milhões de URLs, (Ma et al, 2009b) obteve mais de um milhão de características baseadas no host. O uso exclusivo de características de endereço IP também foi considerado (Chiba et al, 2012). Os recursos de endereço IP são, sem dúvida, mais estáveis, pois é difícil obter novos endereços IP para URLs maliciosas continuamente. Devido a esta estabilidade, eles servem como características importantes na detecção de URLs maliciosas.

4. **Outras características:**

   Nos últimos anos, cresceu bastante o número de provedores de serviços de encurtadores de URLs, que fornecem a URL original para ser representada por uma string mais curta. Isso permite o compartilhamento de URLs nas plataformas de redes sociais, como o twitter, onde as URLs originalmente não se enquadram no limite de 140 caracteres de um tweet. Infelizmente, isso também se tornou uma técnica de ofuscamento popular para as URLs maliciosas. Enquanto os provedores de serviços de encurtadores de URLs tentam não gerar URLs curtas para as maliciosas, eles lutam para fazer um trabalho eficaz, pois eles também dependem principalmente de *blacklists* (Maggi et al, 2013; Gupta et al, 2014). Como resultado, uma direção de pesquisa recentemente emergente tornou-se ativa onde as características de contexto da URL são obtidas, ou seja, as características das informações em background onde a URL foi compartilhada. (LEE; KIM, 2012) usam

informações de contexto derivadas dos tweets onde a URL foi compartilhada. (Wang et al, 2013) usou dados de tráfego de clique para classificar URLs curtas como maliciosas ou não. (Cao et al, 2015) propõem outra direção de características para identificar URLs maliciosas - eles também se concentram em URLs compartilhadas nas mídias sociais e visam identificar a natureza maliciosa de uma URL, realizando análises comportamentais dos usuários que as compartilhavam e os usuários que clicaram nelas. Essas características são formalmente chamadas de características "baseadas em postagem" e características "baseadas em clique". (Alshboul et al, 2015) abordam esse problema com uma categorização sistemática de características de contexto que incluem características relacionadas ao conteúdo (propriedades léxicas e estatísticas do tweet), contexto das características do tweet (tempo, relevância e menções dos usuários) e características sociais (seguindo, seguidores , localização, tweets, retweets e contagem favorita).

Algumas outras características utilizadas foram concebidas para medir a popularidade da URL. Uma das primeiras abordagens para a aplicação de técnicas estatísticas para detectar URLs maliciosas (Garera et al, 2007) visava identificar probabilisticamente a importância de características específicas projetadas à mão. Estas incluem características baseadas em página (classificação da página, qualidade, etc.), características baseadas em domínio (presença na tabela de domínio branco), características baseadas em tipo (tipos de ofuscação) e características baseadas em Palavra (presença de palavras-chave como "confirmar", "Bancário", etc.). (Thomas et al, 2011) usam as características baseadas em URL e baseadas em conteúdo e, além disso, gravam a URL inicial, a URL de destino e a cadeia de redirecionamento. Além disso, eles gravam o número de popups e o comportamento dos plugins, que foram comumente usados pelos spammers. (Choi et al, 2011) propuseram o uso de novas categorias de características: Popularidade do Link e Network Features. A popularidade do link é marcada com base em links recebidos de outras páginas da web. Esta informação foi obtida de diferentes motores de busca. Para tornar o uso dessas características robusto para a manipulação, eles também propõem o uso de determinadas métricas que validam a qualidade dos links. Eles também usam uma métrica para detectar links de URL de spam. Para o seu trabalho, eles usam essas características em conjunto com a característica baseado em conteúdo lexical e características baseadas em host. (Eshetet et al, 2013) usaram características de reputação social de URLs seguindo seu compartilhamento público no Facebook e no Twitter. (Stringhini et al, 2013) incorporou informações sobre cadeias de redirecionamento em gráficos de redirecionamento, o que forneceu informações sobre a detecção de URLs maliciosas.

2.9 Aprendizado de máquina

Na ciência da computação, Aprendizado de Máquina é uma subárea da Inteligência Artificial que tem como objetivo estudar e produzir técnicas ou sistemas computacionais capazes

de adquirir conhecimento de forma automática. Um sistema de aprendizado é um programa de computador que toma decisões baseado em experiências acumuladas através da solução bem sucedida de problemas anteriores. Os diversos sistemas de aprendizado de máquina possuem características particulares e comuns que possibilitam sua classificação quanto à linguagem de descrição, modo, paradigma e forma de aprendizado utilizado (RESENDE, 2003).

Aprendizado de Máquina é uma ferramenta poderosa, mas não existe um único algoritmo que apresente um bom desempenho para todos os problemas (DIETTERICH, 1997). Dessa forma, é importante compreender o poder e a limitação dos diferentes algoritmos utilizando alguma metodologia de avaliação que permita comparar algoritmos.

Técnicas de aprendizado 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 e fala, robótica, etc. Neste trabalho o aprendizado de máquina será utilizado para a detecção de URLs maliciosas.

Existem diversos métodos de aprendizado de máquina disponíveis e utilizados com grande efetividade em diversas aplicações. A seleção do melhor classificador depende de uma série de variáveis, dentre elas, o tipo de problema a ser tratado, a natureza e a disponibilidade de dados, o desempenho, entre outras.

Dois dos métodos de aprendizado de máquina mais adotados são:

- **Aprendizado supervisionado:**

  No aprendizado supervisionado, é fornecido ao sistema de aprendizado um conjunto de exemplos com a saída conhecida, ou seja, cada exemplo observado é descrito por um conjunto de atributos e pelo valor da classe à qual o exemplo pertence (RUSSEL; NORVIG, 2002). Neste aprendizado, acontece o seguinte: o programa é treinado sobre um conjunto de dados pré-definidos, aprende o que precisa e toma decisões inteligentes quando recebe novos dados. Depois de treinado, o algoritmo deve ser capaz de prever resultados de novos exemplos e assim aumentar a eficácia cada vez mais com novos exemplos.

- **Aprendizado não supervisionado:**

  No aprendizado não supervisionado, os algoritmos assumem que não se conhece a classe à qual os exemplos pertencem e procuram encontrar nos valores de atributos similaridades ou diferenças que possam, respectivamente, agrupar os exemplos pertencentes à mesma classe ou dispersar os exemplos de classes distintas (RUSSEL; NORVIG, 2002).

Embora o aprendizado supervisionado forneça uma precisão muito melhor, o aprendizado não supervisionado fornece uma abordagem rápida e confiável para derivar conhecimento de um conjunto de dados.

2.10 Algoritmos de aprendizado de máquina para detecção de URLs maliciosas

Existem vários algoritmos de aprendizado de máquina na literatura, que podem ser aplicados para a solução de detecção de URLs maliciosas. Depois de converter URLs em vetores de características, muitos desses algoritmos de aprendizado podem ser geralmente aplicados para treinar um modelo preditivo de forma bastante direta. Nesta seção, vamos discutir sobre esses algoritmos.

- **Naive Bayes:** Considerado o classificador mais utilizado em aprendizado de máquina, o classificador Naive Bayes é uma técnica simples bastante aplicada ao problema de classificação de tráfego de Internet. A principal vantagem de Naive Bayes é a baixa complexidade na fase de treinamento, tendo em vista que essa fase envolve apenas o cálculo de frequências para que as probabilidades sejam obtidas. Essa peculiaridade faz com que Naive Bayes seja indicado para aplicações onde o treinamento precisa ocorrer de forma online e com frequência regular.

- **SVM:** É uma técnica de classificação amplamente aplicada em detecção e classificação de URLs, fundamentada nos princípios da Minimização do Risco Estrutural (*Structural Risk Minimization - SRM*). Sua finalidade é buscar minimizar o erro com relação ao conjunto de treinamento (risco empírico), assim como o erro com relação ao conjunto de teste, isto é, conjunto de amostras não empregadas no treinamento do classificador (risco na generalização). O objetivo de SVM consiste em obter um equilíbrio entre esses erros, minimizando o excesso de ajustes com respeito às amostras de treinamento (*overfitting*) e aumentando consequentemente a capacidade de generalização. As vantagens desse classificador são: conseguir lidar bem com grandes conjuntos, possuir um processo de classificação rápido e possuir uma baixa probabilidade de erros de generalização. A desvantagens são: precisa definir um bom kernel (função que define a estrutura do espaço de características onde o hiperplano de separação ótima será encontrado) e empregar um tempo de treinamento longo, dependendo do número de dimensionalidade dos dados.

- **KNN:** É um algoritmo de classificação baseado no vizinho mais próximo, ou seja, depende de medidas de distância usadas para classificar objetos com base em exemplos de treinamento, que estão mais próximos no espaço de caraterísticas. A principal vantagem do classificador KNN é a de ser uma técnica simples e facilmente implementada. Já como desvantagem, o uso de poucas instâncias de treinamento pode gerar resultados errados, já que por padrão o KNN intuitivamente usa mais do que um vizinho mais próximo;

- **Árvore de Decisão:** É uma técnica de aprendizado de máquina composta por três elementos básicos: nó raiz, que corresponde ao nó de decisão inicial; arestas, que correspondem às diferentes características; nó folha, que corresponde a um nó de resposta, contendo a classe

à qual pertence o objeto a ser classificado. A principal vantagem do uso de Árvores de Decisão é a de obter regras que explicam claramente o processo de aprendizado, podendo ser usadas para uma compreensão mais completa dos dados e dos atributos mais relevantes para o problema de classificação.

2.11 Detecção através do aprendizado de máquina

No contexto de phishing e de malware, o propósito de um classificador é prever se uma URL de entrada é ou não uma URL maliciosa, com base em suas características (Abu-Nimeh et al, 2007). Inicialmente, um conjunto de características são extraídas da URL que, quando observadas, fornecem evidências para dizer se a URL é maliciosa ou não. Para realizar uma previsão informada, um classificador requer uma fase de treinamento onde aprenda as características que pertencem a cada categoria. Duas das maneiras pelas quais isso pode ser feito são através de um aprendizado supervisionado e não supervisionado.

O aprendizado supervisionada é realizado fornecendo ao classificador vetores de características que já estão rotulados com a categoria correta (Mohri et al, 2012). A abordagem não supervisionada, por outro lado, envolve o fornecimento de vetores de características não rotulados, e o classificador infere as categorias através de processos como agrupamento (Davis et al, 2005). Uma vez que a fase de aprendizado ocorreu, o classificador poderá categorizar as entradas não vistas.

Para escolher o algoritmo de classificação que irá utilizar, depende da natureza dos dados sendo classificados, bem como dos requisitos comportamentais. É evidente que diferentes classificadores terão desempenho preditivo diferente, conforme investigado por (Abu-Nimeh et al, 2007). No contexto da classificação de URLs maliciosas, um desempenho preditivo inadequado pode levar a bloqueios de URLs benignas ou URLs maliciosas que não estão na lista negra.

2.12 Métricas de avaliação de classificação

O processo de otimização de um classificador pode ser visto como um problema de maximização de várias métricas de avaliação. Inicialmente, os resultados da classificação envolvendo n classes podem ser representados por uma matriz de confusão $n \times n$ (Zander et al, 2006). A matriz de confusão (Figura 4) para um problema binário permite a extração de métricas, como acurácia, precisão e Revocação (Batista et al, 2004), conforme mostrado abaixo.

2.12.1 Acurácia (Accuracy) (Batista et al, 2004)

A acurácia representa a proporção de amostras que foram classificadas corretamente e podem ser calculadas como:

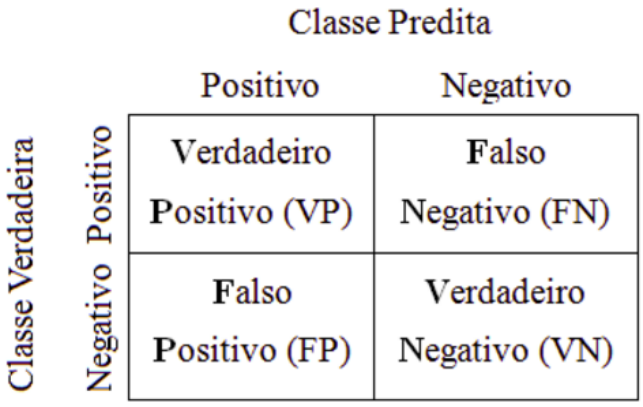

Figura 4: Matriz de Confusão

$$\frac{VP + VN}{VP + FN + FP + VN}$$

2.12.2 Taxa de erro (Error Rate) (Batista et al, 2004)

A taxa de erro é o inverso da acurácia e representa a proporção de amostras que foram classificadas de forma incorreta e é calculada como:

$$\frac{FN + FP}{VP + FN + FP + VN}$$

2.12.3 Precisão (Precision) (Zander et al, 2006)

A precisão mede o número de amostras rotuladas como positivas que de fato são positivas. É dado por:

$$\frac{VP}{VP + FP}$$

2.12.4 Revocação (Recall) (Zander et al, 2006)

Recall, também conhecido como True Positive Rate, mede a proporção de amostras positivas que foram corretamente rotuladas como positivas. É calculado como:

$$\frac{VP}{VP + FN}$$

2.12.5 Medida F (F1 Score) (Yang et al, 1999)

A medida F1 é uma média harmônica de precisão e recall, e varia de 0 a 1. É dada por:

$$\frac{2rp}{r+p}$$

Onde r é recall e p é precisão.

2.13 Catálogo de Fraudes da RNP

O Catalogo de Fraudes da RNP foi criado em 2008, com objetivo de coletar fraudes recebidas por e-mail pela população em geral e analisar, filtrar e catalogar essas fraudes, criando um repositório de mensagens conhecidamente fraudulentas e alertando a comunidade sobre como se proteger desse tipo de ataque. Criado pelo Centro de Atendimento a Incidentes de Segurança da RNP (CAIS/RNP) e mantido atualmente em parceria com o Ponto de Presença da RNP na Bahia (PoP-BA/RNP), o Catálogo de Fraudes da RNP é a única fonte de informações aberta e online sobre fraudes eletrônicas no Brasil, sendo bastante utilizado pela população em geral como uma base de conhecimento para validação de mensagens de e-mail suspeitas (BRITO et al., 2015).

2.14 CaUMa

O CaUMa (Catálogo de URLs Maliciosas) é um serviço gratuito e público criado pelo CERT.Bahia, que disponibiliza um meio de consulta a URLs fraudulentas identificadas na Internet. O propósito desse serviço é ajudar a comunidade a se proteger das diversas fraudes que estão circulando no mundo digital. Através do CaUMa é possível verificar se a URL de um site, que apresenta suspeitas de ser uma fraude e/ou um site malicioso, já foi identificada como fraude. Para isso é necessário inserir a URL para análise e assim o resultado irá indicar se a URL encontra-se no banco do CaUMa e das principais *blacklists* como maliciosa.

As URLs que são armazenadas no banco de dados do CaUMa são fraudes direcionadas ao público brasileiro. Essas URLs são coletadas em parceria com o Catálogo de Fraudes da RNP, Banco do Brasil e outros parceiros estratégicos. Apesar de ser um serviço recente, está crescendo rapidamente e conseguindo vários parceiros. No site é possível visualizar alguns dados estatísticos bastante interessantes, como em 5 que mostra a quantidade de fraudes por categoria e 2 que mostra os domínios mais utilizados para a realização desses ataques.

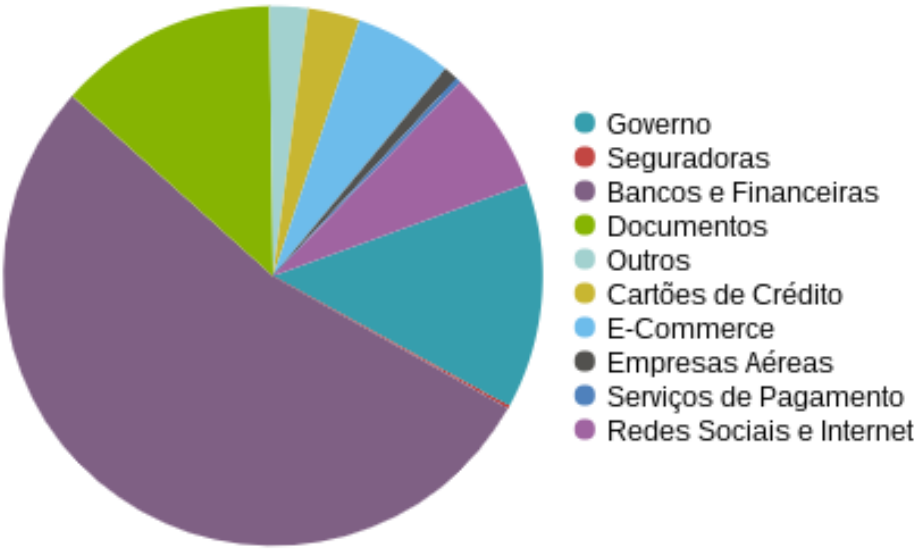

Figura 5: Fraudes por categoria (CERT.Bahia, 2017)

Tabela 2: Top domínios mais utilizados

Ocorrências	**Domínio**
516	www.situacao.cadastral.commbr.com
133	bit.ly
64	googledrive.com
54	atendimentobrasilonlineita.com
50	www.sugarsync.com
44	goo.gl
31	www.123contactform.com
30	dl.dropboxusercontent.com
24	zoomsat.com.br
21	www.dropbox.com

3 Trabalhos Relacionados

Em pesquisas preliminares não foi encontrada nenhuma aplicação, solução ou proposta que seja focada para detecção automática de URLs maliciosas direcionadas ao público brasileiro. Entretanto, foram encontrados trabalhos que utilizam bases de URLs internacionais para extrair as características e treinar uma máquina de aprendizado.

(BEZZERA; FEITOSA, 2015) fizeram uma investigação sobre a capacidade de validação e classificação de URLs como benignas e suspeitas/maliciosas através de determinadas características extraídas das próprias URLs, empregando técnicas de aprendizagem de máquina. Analisando a literatura foi possível enumerar mais de 75 características extraíveis de uma URL que puderam ser aplicadas na sua classificação. Foram desenvolvidos scripts em perl para extração de cada uma das características. Para realizar o estudo, foram utilizadas 20.000 URLs, sendo 10.000 da base DMOZ e as outras 10.000 da base PhishTank. Após a coleta dessas características, foram realizados testes com alguns classificadores de aprendizado de máquina: Naive Bayes, KNN, SVM e Árvore de Decisão (J.48). Com base nos dados obtidos, o classificador J.48 (Árvore de Decisão) foi o mais ajustado, obtendo uma taxa de 95,10% de precisão e 95,11% de taxa de detecção, além de um baixo índice de falso positivo (4,90%), considerando o conjunto de dados fornecidos.

Em (OLIVO, 2010) é feita uma avaliação das características para detecção de phishing de e-mail. Apenas 19 características de phishing foram consideradas importantes para essa pesquisa. O algoritmo de aprendizagem de máquina adotado nessa pesquisa foi o SVM (Support Vector Machines – Máquinas de Vetor de Suporte), pois foi desenvolvido originalmente para resolver esse tipo de problema. A literatura técnica mostra que o SVM tem sido aplicado com bastante sucesso em diversos domínios de aplicação, inclusive na detecção de phishing. Nesse artigo é dito que o software para extração das características foi desenvolvido em shell script.

(Eshetet et al, 2013) propõem uma abordagem que faz uso de busca e otimização evolutiva para integrar com modelos de detecção baseados em aprendizagem para uma análise mais precisa de páginas fraudulentas. Para isso, a abordagem, denominada EINSPECT, inicia com uma população de modelos candidatos treinados usando algoritmos de aprendizagem padrão baseados em características discriminativas extraídos da URL, do código HTML, de código JavaScript e metadados sobre a reputação da página em sites de redes sociais. Em seguida, emprega algoritmos genéticos para automaticamente procurar e otimizar a melhor interação de recursos e algoritmos de aprendizagem. Usando o modelo mais apto, ele detecta páginas Web desconhecidas e as identifica como maliciosas ou benignas.

(Sheng et al, 2007) mostra que *blacklists* não provêem proteção durante o início dos ataques de phishing, pois a URL ou site deve ser inserido na *blacklist* para que essa forma de

proteção comece a funcionar. Além disso, como as URLs utilizadas pelos phishers possuem tempo de vida curto, a utilização de *blacklists* acaba sendo ineficiente. Nessa pesquisa mostra que 63% das campanhas de phishing analisadas pelos autores duraram menos de duas horas. Existem diversas soluções que visam informar ao usuário se uma URL é ou não perigosa, as abordagens baseadas em aprendizagem de máquina vêm ganhando espaço.

(Basnet et al, 2014) descobriram que Random Forest (RF) é o melhor entre 7 outros algoritmos em termos de precisão e velocidade. Além disso, eles adotam apenas a abordagem léxica, eles relatam uma precisão de 85,38% com uma taxa de falsos positivos de 8,22% apenas utilizando as características léxicas, mostrando que apenas analisando a parte léxica é possível conseguir resultados bons de detecção.

(Ma et al, 2009a) exploram o potencial do aprendizado de máquina utilizando características léxicas mais as características baseadas em host. Eles afirmam que as características lexicais tendem a "parecer diferentes", enquanto as características baseadas no host descrevem "onde", "quem" e "como" as URLs são gerenciadas. Seus conjuntos de dados são executados em 3 algoritmos de aprendizado de máquina: Naive Bayes, SVM e Regressão Logística. Eles relatam que, ao executar o conjunto de dados de características lexicais, o uso da Regressão Logística gera uma precisão de 98,07%, enquanto a execução de recursos completos (baseados em léxico e host) usando o mesmo algoritmo gera uma precisão de 98,76%. Mostrando também que apenas utilizando um conjunto de dados com características léxicas já é suficiente para conseguir bons resultados.

Técnicas que fazem uso de características baseadas no HTML para detectar phishing também são aplicadas. (C. Ludl and S. Mcallister and E. Kirda and C. Kruegel, 2007) aplicou um algoritmo de árvore de decisão J48 em 18 características baseadas no HTML e na URL, alcançando um VP de 83,09% e um FP de 0,43% sobre um conjunto de dados com 4149 URLs benignas e 680 URLs maliciosas.

Estudos (Y. Zhang and J. Hong and L. Cranor, 2007) mostraram que os mecanismos de pesquisa e serviços de terceiros, como o WHOIS, são eficazes em fornecer pistas sobre a legitimidade de uma página da Web. Neste trabalho é utilizada essas ferramentas para obtenção de algumas das características.

Também existem trabalhos que tentam combinar os méritos de métodos baseados em blacklists e outras características. Em um trabalho de (G. Xiang and B. A. Pendleton and J. I. Hong and C. P. Rose, 2010) propuseram uma abordagem probabilística baseada em conteúdo, que aproveita as blacklists existentes verificadas pelo homem e aplica a técnica de shingling, um algoritmo popular de detecção que é usado pelos mecanismos de busca, para detectar phishing. Seu algoritmo alcançou 0,03% de FP com um VP de 73,53%.

Além da pesquisa acima, também estão disponíveis barras de ferramentas anti-phishing baseadas em diferentes técnicas, muitas das quais exploram blacklists para garantir um FP baixo.

Produtos bem conhecidos incluem o Microsoft Internet Explorer e o Google Safe Browsing. Sua eficácia foi medida em (Sheng et al, 2009).

4 Metodologia

Neste capítulo apresentamos a metodologia proposta para chegar ao objetivo de criar um método de detecção automática de URLs maliciosas que são direcionadas ao público brasileiro.

4.1 Ambiente de Experimentação

Os experimentos realizados na elaboração desta monografia foram executados em apenas uma máquina. Um notebook com sistema operacional Linux, distribuição Debian 8 64 bits, 8 GB de memória RAM, disco de 1 TB e um processador Intel Core i5. Para a extração das características, foi utilizado um software que foi desenvolvido em Python 3 e para a execução dos algoritmos de classificação e para a análise dos dados foi utilizado o software Weka, na versão 3.6.14.

4.2 Base de URLs

Para elaboração dessa monografia foram utilizadas três bases de URLs:

- **CaUMa - Base de URLs maliciosas (nacionais):**

  A extração das URLs maliciosas nacionais foi realizada a partir da base de dados do CaUMa, na qual tivemos acesso, que é um catálogo de URLs Maliciosas que são direcionadas para o público brasileiro. Nessa base foram coletadas somente as URLs classificadas como Phishing e que ainda estavam online. Foi necessário desenvolver um script em Python para poder realizar a consulta no banco de dados do CaUMa e já trazer as URLs com a formatação correta, pois no banco de dados as URLs estavam guardadas de forma fragmentada, dividindo a URL em protocolo, host, caminho, parâmetros, query e fragmentos.

- **UFBA - Base de URLs benignas (nacionais):**

  A extração das URLs benignas nacionais foi realizada a partir de um Sistema de Detecção de Intrusão configurado em modo espelhamento para algumas VLANs da UFBA. Por questões de privacidade e anonimização dos dados, não é possível revelar as redes que foram utilizadas nesta análise. Além disso, também foi realizado um trabalho manual em cada uma dessas URLs para poder garantir que realmente eram URLs benignas, fazendo uma verificação a olho nu.

- **Fsecurify - Base de URLs benignas e maliciosas (internacionais):**

As URLs internacionais, tanto maliciosas quanto benignas, foram extraídas do repositório <https://github.com/faizann24/Using-machine-learning-to-detect-malicious-URLs>, que disponibiliza um arquivo com 420.465 URLs, sendo que 74.452 são maliciosas e 346.013 são benignas. Esse repositório foi disponibilizado em um artigo sobre detecção de URLs maliciosas utilizando aprendizado de máquina, que foi apresentado pela empresa Fsecurify. No artigo fala como foi que conseguiram essas URLs. O conjunto de dados de URLs maliciosas foi obtido através de uma varredura de vários sites que oferecem links maliciosos, como por exemplo, <vxvault.net>. Já as URLs benignas foram obtidas a partir de um link dado em um trabalho de pesquisa.

Para realizar o treinamento e ajuste dos parâmetros dos classificadores, foram utilizadas 7.112 URLs das bases nacionais, sendo 3.950 oriundas da base CaUMa e 3.162 da base UFBA. Já da base internacional, foram utilizadas 4.000 URLs maliciosas e 3.600 URLs benignas. Na Tabela 3 é possível visualizar alguns exemplos de URLs tiradas dessas bases.

Tabela 3: Exemplos de URLs das bases CaUMa, UFBA e Fsecurify

URL	**Base**
http://sysfiew.com/atualizacaocadastral/index1.php	CaUMa
http://35.227.38.175/171/telas/bb	CaUMa
http://www.tusdelicias.com/img/902384902509-823409859032452890234/cadastro.php	CaUMa
http://bit.do/BB_Atualizaca0	CaUMa
http://www.bb.com.br.app.centralmobile.mobi/1-1-1-1/0_0-1_fisica/session.php	CaUMa
http://cocoon.globo.com/v2/user/logged	UFBA
http://teclim.ufba.br/wp-login.php	UFBA
http://educacao.estadao.com.br/blogs/blog-dos-colegios-santi/educacao-e-tecnologia-o-poder-do-podcast-2	UFBA
http://www.bb.com.br/pbb/pagina-inicial/voce/produtos-e-servicos/ponto-pra-voce/pontos-no-cartao	UFBA
http://f.i.uol.com.br/estudiofolha/images/17338236.jpeg	UFBA
http://apple.security-block.com/Apple%20-%20My%20Apple%20ID.html	Fsecurify (Maliciosa)
http://facebook-tw.zp.ua//pafumokat/bloqyxpn.php	Fsecurify (Maliciosa)
http://insuregem.com/verify/gmail/login.php	Fsecurify (Maliciosa)
http://chamberofcommerce.com/oakland-ca/10647854-lake-temescal-beach-house	Fsecurify (Benigna)
http://spcnet.tv/TVB-Series/The-King-of-Yesterday-and-Tomorrow-review-r345.html	Fsecurify (Benigna)
http://metronews.ca/ottawa/life/article/1031818–cadillac-cts-v-will-haul-something-other-than-your-groceries	Fsecurify (Benigna)

4.3 Construção do dataset

Para poder construir o dataset, foi necessário desenvolver um software, a linguagem escolhida para o desenvolvimento desse software foi o Python 3, pela sua simplicidade, facilidade de uso e por ser uma linguagem que já possuo experiência. O funcionamento desse software consiste nas seguintes etapas:

1. O software recebe como entrada dois argumentos, o primeiro é o arquivo que contém as URLs e o segundo é o nome do arquivo de saída a ser criado.
2. Cria-se o arquivo de saída e então preenche a primeira linha com 118 colunas que são separadas por vírgula. Essas colunas contêm o nome de cada característica e a última coluna é onde diz se é maliciosa ou não.
3. É iniciado um loop para poder pegar cada URL do arquivo de entrada, extrair as 117 características e escrever no arquivo de saída nas colunas correspondentes. Para cada característica existe uma função que é executada para poder extraí-la, essas características podem depender de conexão com a Internet ou não. O tipo do valor de retorno depende da característica, podendo ser valor binário, inteiro, float, string, etc. É importante ressaltar que todos os vetores de características foram preenchidos com interrogação (?) quando informações não puderam ser extraídas da URL. Isso ocorre somente para características que dependem de conexão e outros serviços.
4. Após o fim do loop, o arquivo do dataset é gerado, já estruturado e no formato CSV (Comma-Separated Values).

A execução do software nas bases de URLs durou cerca de 3 dias. O software juntamente com o dataset estão públicos e podem ser encontrados em <https://github.com/lucasayres/url-feature-extractor>.

Após analisar o dataset gerado, percebemos que algumas características possuíam muitos valores faltantes (missings), isso acontece devido a algumas características que dependem de conexão com a internet ou que depende da URL ainda estar online, o tratamento destes casos é necessário para que os resultados sejam confiáveis. Para realizar o tratamento dos valores faltantes (missings), representados pela interrogação (?) no dataset, foi necessário utilizar o método de imputação pela média ou moda, onde foi aplicado os valores da média para os atributos numéricos e valores da moda para os atributos do tipo nominal.

Também foi necessário remover algumas características do dataset ao decorrer dos experimentos, como por exemplo, as características baseadas em blacklist, pois utilizar essas características no treinamento do modelo seria meio que uma forma de trapaça, pois não seria mérito do modelo, e sim das blacklists. Além de serem estatísticas fornecidas por terceiros, não temos controle sobre a qualidade e confiabilidade dos serviços e dados fornecidos por eles.

4.4 Características

Nesta seção apresentamos as características que foram selecionadas para serem extraídas das URLs. A maioria dessas características foram retiradas da literatura, mas também foram adicionadas algumas características extras para ver o grau de importância delas para a detecção das URLs maliciosas. Por questões de implementação, as características foram categorizadas em Léxicas, *Blacklist*, Host e Outras, totalizando 117 características. A seguir é possível ver mais detalhes sobre cada característica que foi implementada.

4.4.1 Características Léxicas

Características léxicas são recursos obtidos com base nas propriedades do nome da URL. Essas características são extraídas através de tokens (símbolos ou palavras-chave) da URL e então é feito algum tipo de contabilização. A seguir estão descritas todas as características léxicas que foram utilizadas nessa monografia.

1. **Quantidade de Tokens na URL, Domínio, Diretório, Arquivo e Parâmetros:** Os tokens considerados foram: “.”, “-”, “_”, “/”, “?”, “=”, “@”, “&”, “!”, “ ”, “ ”, “,”, “+”, “*”, “#”, “$” e “%”. Uma quantidade incomum de tokens pode indicar a presença de uma URL maliciosa.

2. **Comprimento da URL, Domínio, Diretório, Arquivo e Parâmetros:** Essa característica refere-se a medição dos segmentos de uma URL. Existem URLs que possuem uma grande quantidade de caracteres, que diverge do número de caracteres de URLs benignas, podendo indicar a presença de uma URL maliciosa.

3. **Quantidade de TLD (Top Level Domain) na URL:** As URLs benignas normalmente possuem um único TLD. Portanto, padrões de muitos TLDs em uma URL significam que é um site fraudulento (Mohammad et al, 2014). Então caso tenha mais de um TLD na URL, a URL é considerada como maliciosa.

4. **Quantidade de vogais no Domínio:** Essa característica verifica a frequência de vogais no domínio de uma URL. Os autores que utilizaram essa característica em seus testes, falam que as URLs maliciosas tendem a usar menos vogais nos nomes de seus domínios (MCGRATH; GUPTA, 2008).

5. **Domínio da URL em formato de endereço IP:** Verifica se o domínio da URL está no formato de endereço IP. Alguns ataques de phishing utilizam máquinas sem nenhuma entrada DNS, o único modo de referenciá-las é através do endereço IP.

6. **Domínio contém as palavas-chave server ou client:** Verifica a existência dessas palavras no domínio, pois palavras chave como server e client são comuns em ataques de phishing.

7. **Presença de TLD (Top Level Domain) nos Parâmetros da URL:** URLs que incluem outra URL como parâmetro podem ser utilizadas como uma forma de ataque, que visa enganar o usuário redirecionando-o para páginas falsas. Esse tipo de ataque é muito utilizado pelos phishers, já que a URL malicioosa fica mascarada. Um exemplo de URL que pode ser utilizado para enganar o usuário e redirecionar para uma página falsa: http://www.site.tld/index.html?return=http://maliciosa.tld/instalarMalware.exe

8. **Quantidade de Parâmetros:** Essa característica serve para identificar quantos parâmetros existem no argumento da URL.

9. **E-mail presente na URL:** Essa característica procura identificar se existe algum e-mail contido na URL. Na literatura, mostram que os phishers inserem e-mails nas URLs para simular o acesso a algum serviço (LIN, 2008).

10. **Extensão do arquivo:** Essa característica visa identificar a extensão do arquivo na URL, para poder saber quais as extensões que os phishers mais utilizam em seus ataques.

4.4.2 Características baseadas em *blacklist*

3 provedores diferentes de serviços de *blacklist* foram utilizados para estas características, os serviços utilizados foram:

- **Google Safebrowsing:** É um serviço do Google que permite verificar URLs contra as listas constantemente atualizadas do Google de recursos inseguros da Web. Exemplos de recursos inseguros da web são sites de engenharia social (sites de *phishing*) e sites que hospedam *malware* ou softwares indesejados. *Browsers* como o Firefox, Google Chrome e Safari utilizam esse serviço para alertar os usuários quando as mesmas são acessadas (Google Inc, 2016);

- **Phishtank:** É um site gratuito onde qualquer pessoa pode enviar, verificar, rastrear e compartilhar dados de *phishing*. É baseado no serviço *anti-phishing*, usado por empresas como Opera, WOT e YahooMail, para verificar se uma URL é considerada *phishing* (OpenDNS, 2016);

- **WoT:** É uma extensão de navegador que ajuda o usuário a verificar a reputação e informações de segurança de qualquer site na internet, com base na experiência dos usuários (Sami Tolvanen and Timo Ala-Kleemola, 2007);

A seguir estão descritas todas as características baseadas em *blacklist* que foram utilizadas nessa monografia.

1. **Presença da URL, IP e Domínio em *blacklists*:** Essa característica utiliza as APIs do Google Safe Browsing, Phishtank e WoT para poder consultar suas respectivas bases de

dados e então descobrir se a URL pertence a alguma *blacklist*, Caso a URL seja encontrada em alguma *blacklist* retorna informando que é uma URL maliciosa, caso contrário, retorna informando que é uma URL benigna.

4.4.3 Características baseadas em host

As características baseadas em host são obtidas das propriedades do nome do host da URL analisada (Ma et al, 2009b). Com elas, pode-se conhecer a localização dos hosts maliciosos, a identidade, o estilo de gerenciamento e as propriedades desses hosts. Foram escolhidas e implementadas as seguintes características baseadas em host:

1. **Presença do domínio em RBL (Realtime Blackhole List):** Essa característica tem como objetivo identificar se um domínio está presente em uma RBL. Realtime Blackhole Lists (RBLs) são listas da Internet na qual vários serviços de antispam realizam consultas.

2. **Tempo de pesquisa (resposta) de domínio (lookup):** Sites benignos tendem a ser mais acessados, seu tempo de resposta a uma consulta de domínio tende também a ser mais rápida do que de sites maliciosos (Choi et al, 2011).

3. **Registros SPF:** Um registro SPF (*Sender Policy Framework*) para os endereços IP do domínio evitam alguns tipos de ataques de falsificação de e-mail, como phishing e spam.

4. **Localização geográfica do IP:** Tal como acontece com as propriedades do endereço IP, os servidores de domínio com atividades maliciosas podem ser concentrados em regiões geográficas específicas.

5. **Número AS (ou ASN):** Geralmente, todos os endereços IP correspondentes a um nome de domínio coexistem no mesmo ASN porque uma única entidade administrativa mantém o domínio. No entanto, isso não é verdade para domínios maliciosos, pois os exércitos de bot de máquinas hospedeiras comprometidas geralmente pertencem a várias ASNs. Assim, se os endereços IP resolvidos pertencem a muitas ASN, indica que é phishing.

6. **Domínio possui registro PTR:** O registro PTR permite pesquisas de DNS reverso. A existência de um registro PTR é um indicador confiável de que o nome de domínio está bem estabelecido (MA et al, 2011); como tal, o registro PTR é uma característica potencialmente útil.

7. **Tempo (em dias) de ativação do domínio:** Sites de phishing são criados apenas com o intuito de roubar informações dos usuários, e para isso são criados domínios de curta duração (Ma et al, 2009a). Essa característica visa obter o tempo (em dias) em que o domínio está ativo. Quanto mais próxima da data em que um nome de domínio foi registrado, maior a possibilidade de ser um site de phishing.

8. **Tempo (em dias) de expiração do domínio:** Com base no fato de um site de phishing viver por um curto período de tempo, acreditamos que os domínios confiáveis são regularmente pagos por vários anos de antecedência, então se a data restante válida de um nome de domínio for muito curta, é provável que seja um site de phishing.

9. **Número de IPs resolvidos:** Essa característica tem como objetivo contabilizar todos os endereços IP associados ao domínio da URL. URLs benignas abrangem um espaço de endereços IP mais amplo e, portanto, têm mais IPs resolvidos em comparação com um URLs maliciosas.

10. **Número de servidores de nome resolvidos (NameServers – NS):** Essa característica tem como objetivo contabilizar todos os servidores de nome associados ao domínio da URL. Normalmente, os provedores de serviços que hospedam URLs maliciosas compõem um número limitado de servidores de nome.

11. **Número de servidores MX:** O phisher pode configurar os registros MX (Mail eXchange) para criar novos nomes de domínio que podem ser usados para o e-mail.

12. **Valor do Time-to-live (TTL) associado ao domínio:** Cada registro de DNS tem um tempo de vida (TTL) que especifica quanto tempo a resposta correspondente para um domínio deve ser armazenada em cache, para que tanto os clientes DNS quanto os servidores de nomes possam se beneficiar dos efeitos do cache do DNS. Os hosts maliciosos utilizam um endereço IP com TTL médio mais curto do que os hosts legítimos, porque os falsificadores desejam evitar serem armazenados em cache. Quanto mais curto for o TTL, mais rápido um host pode mudar seus registros A.

4.4.4 Outras Características

Nesta seção estão as características que não se enquadram em nenhuma das outras categorias. Foram escolhidas e implementadas as seguintes características:

1. **HTTPS (Hyper Text Transfer Protocol with Secure Sockets Layer):** A existência de HTTPS é muito importante para dar a impressão de legitimidade do site, mas isso não é claramente suficiente. O ideal é verificar o certificado atribuído com HTTPS, para saber se é um certificado válido. O software utilizado para a extração das características verifica tanto a existência de HTTPS quanto se o certificado é válido.

2. **Quantidade de redirecionamentos:** Saber quantas vezes um site foi redirecionado, ajuda a distinguir sites de phishing de benignos. No nosso conjunto de dados, achamos que sites benignos foram redirecionados uma vez no máximo. Por outro lado, alguns sites de phishing foram redirecionados pelo menos 4 vezes.

3. **URL está indexada no Google:** Essa característica verifica se um site (url inteira) está indexado no Google ou não. Quando um site é indexado pelo Google, ele é exibido nos resultados de pesquisa. Normalmente, os sites de phishing são meramente acessíveis por um curto período de tempo e, como resultado, muitos sites de phishing podem não ser indexados pelo Google.

4. **Domínio está indexado no Google:** Essa característica verifica se um domínio está indexado no Google ou não. Quando um domínio é indexado pelo Google, ele é exibido nos resultados de pesquisa. Normalmente, os sites de phishing são meramente acessíveis por um curto período de tempo e, como resultado, muitos domínios de sites de phishing podem não ser indexados pelo Google.

Ao todo foram implementadas 117 características, como pode ser visto na Tabela 4.

Tabela 4: Características implementadas

Características Léxicas		
qtd_ponto_url	qtd_hifen_url	qtd_underline_url
qtd_barra_url	qtd_interrogacao_url	qtd_igual_url
qtd_arroba_url	qtd_comercial_url	qtd_exclamacao_url
qtd_espaco_url	qtd_til_url	qtd_virgula_url
qtd_mais_url	qtd_asterisco_url	qtd_hashtag_url
qtd_cifrao_url	qtd_porcento_url	qtd_tld_url
comprimento_url	qtd_ponto_dominio	qtd_hifen_dominio
qtd_underline_dominio	qtd_barra_dominio	qtd_interrogacao_dominio
qtd_igual_dominio	qtd_arroba_dominio	qtd_comercial_dominio
qtd_exclamacao_dominio	qtd_espaco_dominio	qtd_til_dominio
qtd_virgula_dominio	qtd_mais_dominio	qtd_asterisco_dominio
qtd_hashtag_dominio	qtd_cifrao_dominio	qtd_porcento_dominio
qtd_vogais_dominio	comprimento_dominio	formato_ip_dominio
server_client_dominio	qtd_ponto_diretorio	qtd_hifen_diretorio
qtd_underline_diretorio	qtd_barra_diretorio	qtd_interrogacao_diretorio
qtd_igual_diretorio	qtd_arroba_diretorio	qtd_comercial_diretorio
qtd_exclamacao_diretorio	qtd_espaco_diretorio	qtd_til_diretorio
qtd_virgula_diretorio	qtd_mais_diretorio	qtd_asterisco_diretorio
qtd_hashtag_diretorio	qtd_cifrao_diretorio	qtd_porcento_diretorio
comprimento_diretorio	qtd_ponto_arquivo	qtd_hifen_arquivo
qtd_underline_arquivo	qtd_barra_arquivo	qtd_interrogacao_arquivo
qtd_igual_arquivo	qtd_arroba_arquivo	qtd_comercial_arquivo
qtd_exclamacao_arquivo	qtd_espaco_arquivo	qtd_til_arquivo
qtd_virgula_arquivo	qtd_mais_arquivo	qtd_asterisco_arquivo
qtd_hashtag_arquivo	qtd_cifrao_arquivo	qtd_porcento_arquivo
comprimento_arquivo	qtd_ponto_parametros	qtd_hifen_parametros
qtd_underline_parametros	qtd_barra_parametros	qtd_interrogacao_parametros
qtd_igual_parametros	qtd_arroba_parametros	qtd_comercial_parametros
qtd_exclamacao_parametros	qtd_espaco_parametros	qtd_til_parametros
qtd_virgula_parametros	qtd_mais_parametros	qtd_asterisco_parametros
qtd_hashtag_parametros	qtd_cifrao_parametros	qtd_porcento_parametros
comprimento_parametros	presenca_tld_argumentos	qtd_parametros
email_na_url	extensao_arquivo	
Características baseadas em *blacklist*		
url_presente_em_blacklists	presenca_ip_blacklists	dominio_presente_em_blacklists
Características baseadas em host		
dominio_presente_em_rbl	tempo_de_resposta	possui_spf
localizacao_geografica_ip	numero_as_ip	ptr_ip
tempo_ativacao_dominio	tempo_expiracao_dominio	qtd_ip_resolvido
qtd_nameservers	qtd_servidores_mx	valor_ttl_associado
Outras Características		
certificado_tls_ssl	qtd_redirecionamentos	url_indexada_no_google
dominio_indexado_no_google		

4.5 Classificadores

Nesta seção apresentamos os classificadores que foram selecionados para realizar os experimentos, as métricas de avaliação e os ajustes realizados em cada classificador.

4.5.1 Classificadores selecionados

Os classificadores selecionados para realizar os experimentos foram: Naive Bayes, KNN, SVM e Árvore de Decisão (J48). Essa escolha foi feita com base na literatura, onde mostra que esses classificadores são os mais utilizados e que possuem um melhor resultado quando se trata de detecção de phishing. A maioria das pesquisas de detecção de fraudes se baseiam na estratégia de aprendizado supervisionado, pois gera melhores resultados além de permitir a criação de um modelo preditivo para identificação de futuras fraudes. A Seção 2.10 explica cada um desses classificadores de forma detalhada.

4.5.2 Métricas de Avaliação

Para a medição dos resultados, foi utilizada a técnica de validação cruzada (*cross-validation*), utilizando o método k-fold, com 10 partições. O método de validação cruzada consiste em dividir o conjunto de dados em dez partes iguais e testar dez vezes, onde em cada teste uma parte é usada no conjunto de teste e as outras nove são usadas no conjunto de treino. Após a execução dos dez testes, o resultado das medidas de qualidade são uma média entre o resultado de todas as dez execuções. Mantendo-se assim a mesma proporção em todos os experimentos a fim de permitir a comparação dos resultados obtidos.

As medidas utilizadas para a análise de desempenho foram: Revocação (*Recall*), Acurácia (*Accuracy*), Precisão (*Precision*) e F1 Score.

4.5.3 Verificando se modelos treinados com bases internacionais funcionam bem nos dados nacionais

Antes de iniciar os experimentos com as bases nacionais, resolvemos verificar se os modelos treinados com as bases internacionais funcionam bem quando testados com conjuntos de URLs nacionais, para poder comprovar o que já foi dito antes, que os métodos de detecção de URLs maliciosas que existem atualmente não são eficazes quando usados com URLs nacionais, por se tratar de métodos focados em URLs internacionais.

Para a realização desta tarefa, foi necessário abrir o software Weka e carregar o arquivo CSV contendo o conjunto de dados obtidos da base internacional. Na Figura 6, é possível ver uma representação da tela com o conjunto de dados já aberto. Também é possível ver todos os atributos, inclusive a classe. Além dos atributos, é possível visualizar alguns dados estatísticos como o valor máximo e mínimo, média e desvio padrão.

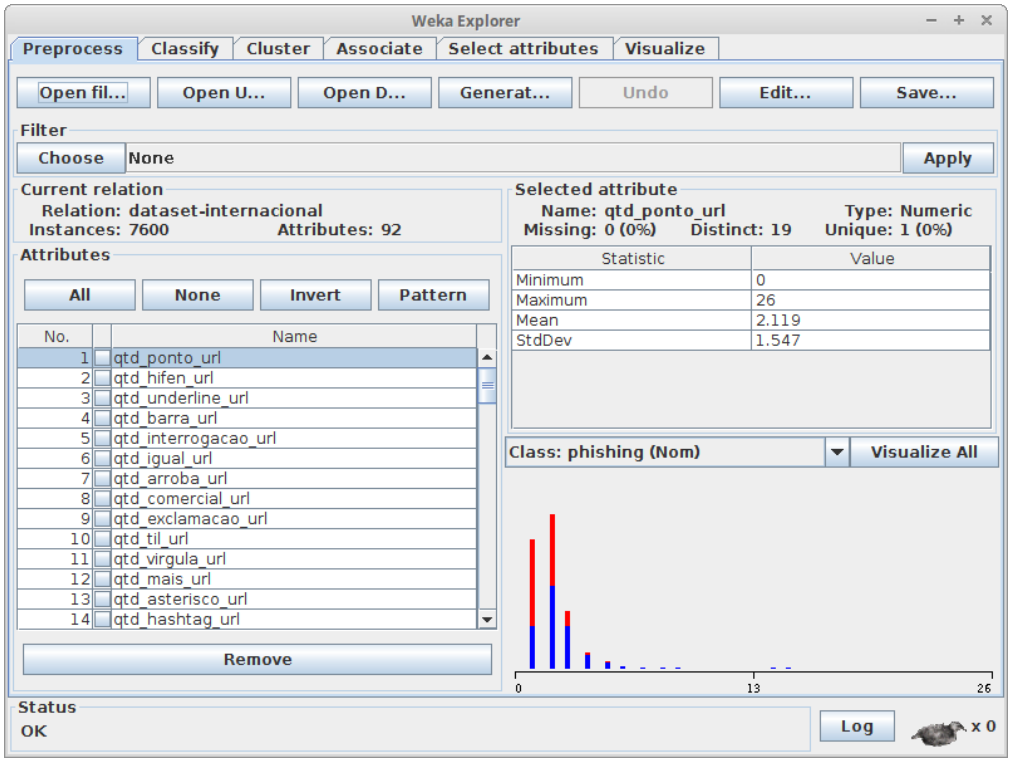

Figura 6: Atributos do conjunto internacional e algumas estatísticas

Após o carregamento do conjunto de dados, foram executados 4 algoritmos (Naive Bayes, KNN, SVM e Árvore de Decisão (J48)) para realizar o treinamento e teste do modelo. A Figura 7 exibe a aba de classificadores (Classify), com o algoritmo J48 selecionado, treinamento utilizando a opção "Use training set", que utiliza o mesmo dataset para treinar e testar o modelo, e a árvore de decisão utilizando a estrutura de representação do próprio Weka.

Terminando a criação dos modelos, foi realizado a exportação dos mesmos, objetivando serem utilizados com o dataset de URLs nacionais. Então foi necessário abrir novamente o Weka, carregar os modelos treinados com a base internacional e então executar o conjunto de dados nacionais no modelo, para que o Weka pudesse trazer o resultado de como o modelo se saiu diante desses dados nacionais.

A partir dos resultados e análises desse experimento que são detalhados na Seção 5.1, é possível visualizar que os modelos treinados com bases internacionais não funcionam muito bem quando são treinados com bases nacionais, mostrando que se faz realmente necessário a criação de um método de detecção de URLs maliciosas direcionadas à comunidade brasileira.

Esse resultado obtido deu uma motivação a mais para poder prosseguir com este trabalho e tentar chegar a um modelo eficaz na detecção de URLs maliciosas direcionadas a comunidade brasileira. Nas seções a seguir são detalhados os experimentos realizados utilizando essas bases de URLs nacionais.

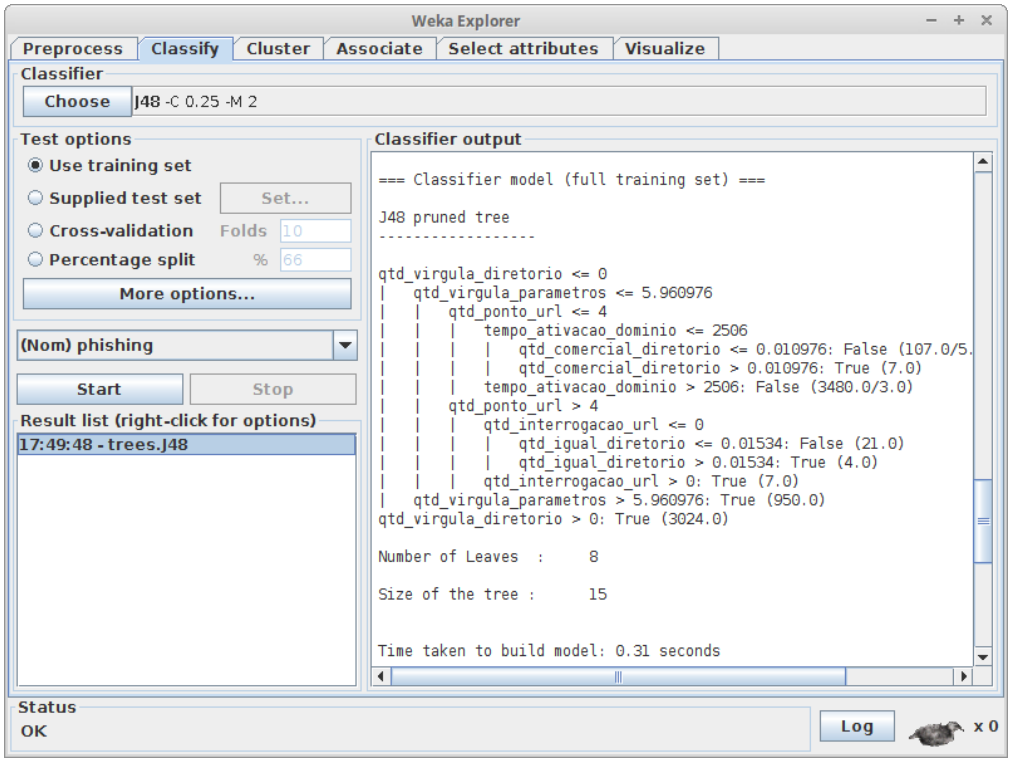

Figura 7: Árvore de decisão utilizando o algoritmo J48

4.5.4 Ajustes dos Classificadores

Foram realizados alguns ajustes nos valores dos principais parâmetros de cada classificador, para poder obter um melhor resultado sobre o conjunto de dados nacional. O único classificador que não foi possível realizar ajustes, foi o Naive Bayes, que não possui parâmetros ajustáveis no Weka.

1. **Classificador baseado em Árvore de Decisão (J48):**

   No classificador J48 foram realizados ajustes no parâmetro de Fator de Confiança, com o objetivo de analisar a precisão das regras geradas, esse fator estabelece a confiança na base de treinamento e na avaliação de erro. Quanto menor o valor do Fator de Confiança, maior será a probabilidade do nó ser podado em função dos nós estáveis, ou seja, menor será o tamanho da árvore e a quantidade de nós que poderiam levar a erros de classificação. O valor padrão do Fator de Confiança no WEKA para o J48 é 0,25.

   O resultado e análise encontra-se na Seção 5.2.1.

2. **Classificador KNN:**

   No classificador KNN foi realizado ajustes também no fator de confiança. O valor padrão do Fator de Confiança no WEKA para o classificador KNN é 1,0.

   O resultado e análise encontra-se na Seção 5.2.2.

3. **Classificador SVM:**

   Para o classificador SVM, foi ajustado o parâmetro de regularização ou penalização, denominado parâmetro C, que determina a rigidez do modelo em relação à tolerância a erros. Ao aumentar o valor desse parâmetro, o modelo e torna mais rígido e preciso, porém fica mais custoso na fase de treinamento. Porém, ao diminuir esse valor, o modelo fica mais tolerante a erros e menos rígido. O valor padrão no WEKA para o classificador SVM é 1,0.

   O resultado e análise encontra-se na Seção 5.2.3.

4.5.5 Escolha do melhor classificador

Após aplicado os ajustes dos parâmetros para obter o melhor resultado de cada classificador, foi possível realizar a comparação entre eles, para determinar qual o classificador que melhor se ajusta ao conjunto de treinamento. A medição de desempenho dos classificadores foi realizada com base nas medidas descritas na Seção 4.5.2. A análise juntamente com os resultados, são mostrados na Seção 5.3.

4.6 Análise das características

Nesta seção, analisamos as características do nosso conjunto de dados separadamente, para poder enxergar o grau de importância de cada uma delas e quais os valores que dizem se a URL pode ser phishing ou não.

4.6.1 Características com maior poder preditivo

Para poder avaliar quais são as características mais relevantes, com maior poder preditivo, foi utilizado a métrica de avaliação de atributos ReliefF juntamente como o método de busca do tipo Ranker, implementado pelo Weka. O ReliefF avalia o atributo individualmente de acordo com o seu valor para a classe majoritária entre múltiplas instâncias mais próximas do conjunto de dados fornecidos. Já a busca do tipo Ranker, organiza os atributos em ordem decrescente de acordo com a relevância atribuída pelo ReliefF.

Após executar esses métodos no conjunto de dados, foi gerado um resultado trazendo as características de maior poder preditivo. Esse resultado é possível encontrar na Seção 5.4.1.

4.6.2 Distribuição geográfica de phishings

Para poder saber quais os países que mais hospedam as páginas de phishing, foi necessário carregar o dataset no WEKA para melhor visualizar os valores e tipos de cada característica de forma separada. O resultado foi inserido na Tabela 11 que encontra-se na Seção 5.4.2.

5 Resultados

Nesta seção, são apresentados os resultados obtidos através da execução dos classificadores conforme metodologia descrita na Seção 4.5.

5.1 Resultados obtidos ao testar um modelo treinado com base internacional em uma base nacional

A Tabela 5 mostra os resultados obtidos ao utilizar o modelo treinado na base internacional para testar com a base nacional. Esse experimento foi realizado utilizando os classificadores J48, KNN, Naive Bayes e SVM.

Tabela 5: Resultado dos classificadores ao utilizar o modelo treinado na base internacional para testar com a base nacional

	J48	KNN	Naive Bayes	SVM
Acurácia	45,26%	44,51%	36,75%	44,40%
Precisão	73,20%	50,40%	39,30%	44,10%
Recall	2,30%	6,90%	25,50%	0,40%
F1 Score	4,40%	12,20%	31,00%	0,80%

Ao analisar os resultados, percebe-se que quando treinado o modelo nas bases nacionais, não foi possível obter um bom desempenho, chegando a ter uma taxa de acurácia de no máximo 45,26%, utilizando o classificador J48, que foi o classificador que apresentou o melhor resultado.

5.2 Ajustes dos classificadores

Nesta seção são exibidas as análises e os resultados obtidos ao realizar os ajustes nos classificadores.

5.2.1 Classificador baseado em Árvore de Decisão (J48)

A Tabela 6 mostra o resultado do ajuste do Fator de Confiança que foi aplicado no classificador J48.

Tabela 6: Ajuste do Fator de Confiança para o classificador J48

Fator de Confiança	Acurácia	Precisão	Recall	F1 Score
0,001	94,37%	94,50%	95,50%	95,00%
0,01	94,83%	94,90%	95,90%	95,40%
0,1	95,19%	95,60%	95,70%	95,70%
0,25	95,47%	95,90%	95,90%	95,90%
0,5	95,55%	96,10%	95,90%	96,00%
1,0	95,55%	96,40%	95,60%	96,00%

Analisando a Tabela 6, podemos perceber que o melhor resultado foi o que teve o ajuste no Fator de Confiança de valor 1,0, que apresenta a acurácia (95,55%) igual a outro índice, mas possui a melhor precisão (96,40%).

5.2.2 Classificador KNN

A Tabela 7 mostra o resultado do ajuste do Fator de Confiança que foi aplicado no classificador KNN.

Tabela 7: Ajuste do Fator de Confiança para o classificador KNN

Fator de Confiança	Acurácia	Precisão	Recall	F1 Score
1	96,02%	96,30%	96,50%	96,40%
3	95,26%	95,50%	96,00%	95,70%
5	95,00%	95,10%	95,90%	95,50%
7	94,61%	94,50%	95,90%	95,20%

Analisando a Tabela 7 podemos dizer que o ajuste com melhor resultado foi o Fator de Confiança de valor 1, que conseguiu ficar com porcentagem maior em todas as métricas.

5.2.3 Classificador SVM:

A Tabela 8 mostra o resultado do ajuste do parâmetro de regularização ou penalização que foi aplicado no classificador KNN.

Tabela 8: Ajuste do Parâmetro de regularização ou penalização para o Classificador SVM

Fator de Confiança	Acurácia	Precisão	Recall	F1 Score
0,001	81,62%	77,60%	94,10%	85,00%
0,01	90,86%	90,20%	93,80%	91,90%
0,5	93,84%	94,20%	94,80%	94,50%
1,0	94,17%	94,60%	94,90%	94,80%
3,0	94,36%	94,90%	94,90%	94,90%
5,0	94,55%	95,00%	95,20%	95,10%
10,0	94,55%	95,10%	95,10%	95,10%
50,0	94,61%	95,00%	95,30%	95,20%

Percebe-se na Tabela 8 que com o aumento do parâmetro C, aumenta a porcentagem da acurácia e da precisão. O melhor resultado foi utilizando o valor 50 para o parâmetro C.

5.3 Escolha do melhor classificador

Após aplicado os ajustes dos parâmetros, foi possível realizar a comparação entre os resultados obtidos pelos classificadores, para determinar qual o classificador que melhor se ajusta ao conjunto de treinamento.

A tabela 9 mostra a comparação entre os classificadores J48, KNN, Naive Bayes e SVM:

Analisando o resultado da Tabela 9 foi possível constatar que o classificador Naive Bayes teve um desempenho inferior em relação aos demais, tendo uma taxa de acurácia de 79,17% e

Tabela 9: Comparação entre os classificadores J48, KNN, Naive Bayes e SVM

	J48	KNN	Naive Bayes	SVM
Acurácia	95,55%	96,02%	79,17%	94,61%
Precisão	96,40%	96,30%	74,40%	95,00%
Recall	95,60%	96,50%	95,40%	95,30%
F1 Score	96,00%	96,40%	83,60%	95,20%

taxa de precisão de 74,40%, mostrando que esse classificador possui uma capacidade baixa de aprendizado para esse conjunto de dados. O restantes dos classificadores tiveram um desempenho muito bom e com valores próximos, mas o que se saiu melhor foi o classificador KNN, obtendo uma taxa de acurácia de 96,02% e 96,30% de taxa de precisão.

5.4 Análise das características

Nesta seção encontram-se os resultados obtidos a partir da análise feita em cada uma das características de forma separada.

5.4.1 Características com maior poder preditivo

As top 10 características com maior poder preditivo são mostradas na Tabela 10 em ordem decrescente de importância.

Tabela 10: Características avaliadas individualmente de acordo com o critério Relief e classificadas pelo Ranker

Peso	Característica
0.38	tempo de ativação do domínio
0.355	valor ttl associado
0.277	tempo de resposta
0.264	comprimento da url
0.239	tempo de expiração do domínio
0.238	comprimento do arquivo
0.226	quantidade de redirecionamentos
0.214	extensão do arquivo
0.211	comprimento dos parâmetros
0.21	quantidade de parâmetros

A Tabela 10 mostra que a característica “tempo de ativação do domínio” (característica numérica, que possui os valores representados em dias) foi a que conseguiu se sair melhor no conjunto de dados.

As características que se saíram melhor são as que foram obtidas das propriedades do nome do host da URL, como: tempo de ativação do domínio, valor ttl associado e tempo de resposta. Mostrando a importância dessa categoria na detecção das URLs maliciosas.

5.4.2 Distribuição geográfica de phishings

Na Tabela 11 são exibidos os 10 países que mais hospedam páginas de phishing. Essa Tabela mostra que algumas áreas são altamente associadas à atividade de phishing. Enquanto os

Estados Unidos abrigam o maior número de páginas de phishing, o Brasil vem em segundo lugar e em seguida os países do centro e sul Europeu.

Tabela 11: Países que hospedam a maioria das páginas de phishing

País	URLs de Phishing	URLs Benignas	% Phishing
Estados Unidos	2518	1611	63,74%
Brasil	358	1288	9,06%
Canadá	189	65	4,78%
Itália	88	2	2,22%
Alemanha	82	13	2,07%
Holanda	73	6	1,84%
França	61	22	1,54%
Rússia	48	5	1,21%
Reino Unido	46	24	1,16%
Polônia	36	2	0,91%

5.5 Discussão

Através desses experimentos, a análise dos resultados demonstram que o método proposto atingiu o objetivo, que é a detecção de URLs maliciosas direcionadas a comunidade brasileira. De acordo com a Tabela 9, o classificador KNN mantém uma alta acurácia e precisão, com o conjunto de dados utilizado. Comparado com outros trabalhos bem sucedidos [(BEZZERA; FEITOSA, 2015), (Basnet et al, 2014)], o método mostra um desempenho com taxas de precisão e acurácia similares ou superiores a maiorias desses trabalhos. Lembrando que essa comparação está sendo feita com trabalhos que utilizam bases internacionais, pois não foi encontrado nenhum trabalho que utilize algum tipo de base de URLs nacionais.

O resultado da tabela 9 sugere que é possível gerar um modelo eficaz para o cenário nacional, usando um conjunto de dados limitado. Nos experimentos realizados, os conjuntos de dados para as avaliações são subamostrados aleatoriamente para simular as diferentes características. Como esse conjunto de dados foi coletado a partir dos dados reais de phishing, e contendo diferentes características presentes nas URLs, os resultados da experiência podem refletir um cenário anti-phishing mais próximo da realidade nacional.

Nos resultados obtidos algumas URLs de Phishing foram classificadas como benignas, por estarem bem camufladas, ficando online por um longo período de tempo, hospedadas gratuitamente em serviços de hospedagem legítimos, sem possuir palavras-chave relacionadas a phishing, sem erros de português nos textos e em alguns casos, essas URLs estavam em resultados de pesquisa do Google. Acredita-se que, olhando e analisando o conteúdo da página, alguns desses falsos negativos podem ser reduzidos.

É necessário que as características utilizadas no treinamento sejam sempre revistas e atualizadas, pois os atacantes estão criando páginas de phishing cada vez mais difíceis de serem detectadas, utilizando novos recursos de camuflagem, como: serviços de encurtamento de URLs, comprando domínios já existentes com maior tempo de vida, aplicando SEO (Search Engine

Optimization) nas páginas para poder ser melhor rankeado nas buscas do Google, entre outros métodos.

Como o foco desse trabalho é na URL, esse modelo pode ser aplicado em qualquer lugar que uma URL possa ser incorporada, como no e-mail, páginas da web, chat, etc. Pode ser desenvolvido por exemplo, um software de detecção de URLs maliciosas, aplicando as regras geradas pela árvore de decisão do algorítmo J48.

6 Conclusão

Neste trabalho, foi proposto um método de detecção de URLs maliciosas direcionadas a comunidade brasileira, utilizando aprendizado de máquina. Essa proposta tinha como objetivo reduzir a ineficiência dos métodos de detecção existentes, tratando-se de URLs que são direcionadas a população brasileira.

Primeiramente, os trabalhos relacionados foram apresentados visando mostrar quais as características e classificadores mais utilizados. Em seguida, foi realizado um estudo sobre as características para poder definir quais seriam utilizadas, posteriormente agrupadas e apresentadas de forma detalhada neste trabalho.

Para realizar a extração dessas características foi necessário desenvolver um software para poder automatizar esse processo. Sua implementação e funcionamento foram descritos na Seção 4.3. Também foram apresentadas as bases de dados, o processo de construção do dataset, os classificadores selecionados e seus devidos ajustes de configuração, bem como as métricas utilizadas para a avaliação, entre outros dados.

Para alcançar o objetivo proposto neste trabalho, o modelo foi avaliado em conjuntos de dados nacionais reais, comparando os resultados de desempenho dos classificadores J48, KNN, Naive Bayes e SVM. Os resultados obtidos nos experimentos mostraram que a solução anti-phishing proposta foi capaz de detectar URLs de phishing com uma acurácia e precisão de mais de 96%.

A maioria dos classificadores, exceto o Naive Bayes, mostrou estatisticamente métricas de desempenho semelhantes. Para o problema exposto, o classificador KNN, no entanto, proporcionou o melhor desempenho durante os experimentos, mesmo após realizar os ajustes nas configurações de cada classificador para poder obter o melhor resultado de cada um.

Neste trabalho tivemos também algumas contribuições:

- **Características:** Conjunto de características que foi utilizado nos experimentos, explicando detalhadamente sobre cada uma delas e a importância na detecção das URLs maliciosas.
- **Software de extração:** Software onde é possível extrair as características das URLs. Esse software está disponível em <https://github.com/lucasayres/url-feature-extractor>.
- **Bases de dados:** Também foi disponibilizado em <https://github.com/lucasayres/url-feature-extractor> as bases de URLs que foram utilizadas nos experimentos.
- **Dataset:** O dataset contendo as características extraídas também está disponível em <https://github.com/lucasayres/url-feature-extractor>.

- **Análise das características:** Foram realizadas análises das características individualmente, para poder disponibilizar algumas estatísticas e mostrar o grau de importância de cada uma na detecção das URLs maliciosas.
- **Análise dos modelos treinados com bases internacionais em dados nacionais:** Foi disponibilizado uma análise dos modelos treinados com bases internacionais em dados nacionais, mostrando que os modelos internacionais não funcionam bem quando treinados com um conjunto de dados nacionais.
- **Análise de desempenho dos classificadores:** Foram apresentados os resultados de cada classificador e a comparação entre eles.

6.1 Trabalhos futuros

Abaixo estão listados os potenciais trabalhos futuros para dar continuidade a esta pesquisa. Neste sentido, os trabalhos futuros foram divididos da seguinte forma:

- Realizar novos estudos para identificar outras características relevantes que contribuam para a detecção de URLs maliciosas.
- Uso de novas bases de dados.
- Uso de outros classificadores para obtenção de novos resultados.
- Desenvolver um plugin para browsers, que irá analisar a URL visitada e alertar o usuário se é uma URL maliciosa.
- Desenvolver um plugin para clientes de E-mail, que irá analisar as URLs encontradas no corpo do e-mail e então classificar como maliciosa ou benigna, evitando com que os usuários caiam em golpes.
- Disparar automaticamente e-mail para a instituição relacionada com a fraude, alertando-a.

Referências

Abu-Nimeh et al. *A comparison of machine learning techniques for phishing detection*. [S.l.]: Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit, 2007. 60-69 p. Citado na página 33.

ALECRIM, E. *Malwares: O que são e como agem*. 2017. 42-49 p. Disponível em: <https://www.infowester.com/malwares.php>. Citado na página 15.

Alshboul et al, Y. *Detecting malicious short urls on twitter*. 2015. Citado na página 30.

AO Kaspersky Lab. *Spam and phishing in Q1 2016*. 2016. Disponível em: <https://securelist.com/spam-and-phishing-in-q1-2016/74682>. Citado 3 vezes nas páginas 9, 20 e 21.

Basnet et al, B. *LEARNING TO DETECT PHISHING URLs*. IJRET: International Journal of Research in Engineering and Technology, 2014. Disponível em: <http://esatjournals.net/ijret/2014v03/i06/IJRET20140306003.pdf>. Citado 4 vezes nas páginas 9, 26, 38 e 55.

Batista et al. *A study of the behavior of several methods for balancing machine learning training data*. [S.l.]: ACM Sigkdd Explorations Newsletter, 2004. 20-29 p. Citado 3 vezes nas páginas 13, 33 e 34.

BERNERS-LEE, T.; FIELDING, R.; MASINTER, L. *Uniform Resource Identifier (URI): Generic Syntax*. IETF, 2005. RFC 3986 (Standard). (Request for Comments, 3986). Disponível em: <http://www.ietf.org/rfc/rfc3986.txt>. Citado na página 18.

BERNERS-LEE, T.; MASINTER, L.; MCCAHILL, M. *Uniform Resource Locators (URL)*. IETF, 1994. RFC 1738 (Proposed Standard). (Request for Comments, 1738). Obsoleted by RFCs 4248, 4266, updated by RFCs 1808, 2368, 2396, 3986. Disponível em: <http://www.ietf.org/rfc/rfc1738.txt>. Citado na página 18.

BEZZERA, M.; FEITOSA, E. *Investigando o uso de Características na Detecção de URLs Maliciosas*. SBSeg, 2015. Disponível em: <http://sbseg2015.univali.br/anais/SBSegCompletos/artigoCompleto08.pdf>. Citado 2 vezes nas páginas 37 e 55.

BRITO, I. et al. *Catálogo de Fraudes da RNP: 7 anos de experiência no tratamento de fraudes eletrônicas brasileiras*. Conferência Integrada ICCyber ICMedia, 2015. 1–5 p. Disponível em: <https://www.pop-ba.rnp.br/pub/Site/Publicacao0004/ICoFCS2015.pdf>. Citado 4 vezes nas páginas 16, 21, 22 e 35.

BRITO, I. et al. *Catálogo de Fraudes e Catálogo de URLs Maliciosas: Identificação e Combate a Fraudes Eletrônicas na Rede Acadêmica Brasileira*. Sexta Conferencia de Directores de Tecnologia de Información, TICAL 2016, 2016. Disponível em: <https://www.pop-ba.rnp.br/pub/Site/Publicacao0005/artigo-tical2016-catfraudes-cauma.pdf>. Citado na página 16.

C. Ludl and S. Mcallister and E. Kirda and C. Kruegel. *On the effectiveness os techniques to detect phishing sites*. 2007. Citado na página 38.

Canali et al, D. *Prophiler: a fast filter for the large-scale detection of malicious web pages*. [S.l.]: 20th international conference on World wide web, 2011. 197–206 p. Citado 2 vezes nas páginas 15 e 25.

Cao et al, C. *Detecting spam urls in social media via behavioral analysis*. [S.l.]: Advances in Information Retrieval, 2015. 703-714 p. Citado na página 30.

CERT.Bahia. *CaUMa*. 2017. Disponível em: <https://cauma.pop-ba.rnp.br>. Citado 2 vezes nas páginas 9 e 36.

CGI.BR. *Cartilha de Segurança para Internet. Comitê Gestor da Internet no Brasil*. 2012. Disponível em: <http://cartilha.cert.br/livro/cartilha-seguranca-internet.pdf>. Citado 2 vezes nas páginas 20 e 22.

Chiba et al, D. *Detecting malicious websites by learning ip address features*. [S.l.]: 12th International Symposium on. IEEE, 2012. 29-39 p. Citado na página 29.

Choi et al, H. *Detecting malicious web links and identifying their attack types*. [S.l.]: 2nd USENIX conference on Web application development, 2011. 11 p. Citado 2 vezes nas páginas 30 e 45.

DAIGLE, L. *WHOIS Protocol Specification*. IETF, 2004. RFC 3912 (Draft Standard). (Request for Comments, 3912). Disponível em: <http://www.ietf.org/rfc/rfc3912.txt>. Citado na página 24.

DANIEL, C. G. *DNS - Um Guia para Administradores de Redes*. [S.l.]: Brasport, 2006. Citado 2 vezes nas páginas 22 e 23.

Davis et al. *Automated traffic classification and application identification using machine learning*. [S.l.]: The IEEE Conference on Local Computer Networks 30th Anniversary, 2005. 250–257 p. Citado na página 33.

DIETTERICH, T. *Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms.* [S.l.]: Neural Computation, 10(7):1895–1924, 1997. Citado na página 31.

DIEZ. *Phishing. Problemática relativa a la calificación jurídica de laparticipación de los denominados "mulerosbancarios". Estado actual de nuestra doctrina y jurisprudência*. 2013. Disponível em: <http://www.elderecho.com/penal/Phising-Problematica-calificacion-participacion-jurisprudencia_11_533680004.html>. Citado na página 20.

Eshetet et al, B. *Binspect: Holistic analysis and detection of malicious web pages*. [S.l.]: UFAM, 2013. 149-166 p. Citado 4 vezes nas páginas 15, 25, 30 e 37.

Finjan Research Center. *Cybercrime Intelligence: Cybercriminals use Trojans & Money Nules to Rob Online Banking Accounts*. 2009. 1–9 p. Citado na página 22.

G. Xiang and B. A. Pendleton and J. I. Hong and C. P. Rose. *A hierarchical adaptive probabilistic approach for zero hour phish detection*. [S.l.]: 15th European Symposium on Research in Computer Security, 2010. 268–285 p. Citado na página 38.

Garera et al, S. *A framework for detection and measurement of phishing attacks*. [S.l.]: ACM workshop on Recurring malcode, 2007. 1-8 p. Citado 2 vezes nas páginas 16 e 30.

Google Inc. *Google Safe Browsing*. 2016. Disponível em: <https://developers.google.com/safe-browsing>. Citado na página 44.

Gupta et al, N. *bit.ly/malicious: Deepdive into short url based e-crime detection*. [S.l.]: APWG Symposium on. IEEE, 2014. 14–24 p. Citado na página 29.

GöRLING, S. *An overview of the Sender Policy Framework (SPF) as an anti-phising mechanism*. Internet Research, 2007. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.670.3345&rep=rep1&type=pdf>. Citado na página 24.

HAWKINSON, J.; BATES, T. *Guidelines for creation, selection, and registration of an Autonomous System (AS)*. IETF, 1996. RFC 1930 (Best Current Practice). (Request for Comments, 1930). Disponível em: <http://www.ietf.org/rfc/rfc1930.txt>. Citado na página 25.

Kaspersky Lab. *Kaspersky security bulletin statistics 2016*. 2016. Disponível em: <https://kasperskycontenthub.com/securelist/files/2016/12/Kaspersky_Security_Bulletin_2016_Statistics_ENG.pdf>. Citado na página 15.

Kaspersky Lab. *Spam and phishing in Q2 2017*. 2017. Disponível em: <https://securelist.com/spam-and-phishing-in-q2-2017/81537>. Citado na página 15.

Kaspersky Lab. *Spam and phishing in Q3 2017*. 2017. Disponível em: <https://securelist.com/spam-and-phishing-in-q3-2017/82901/>. Citado na página 15.

Kolari et al, P. *Svms for the blogosphere: Blog identification and splog detections*. [S.l.]: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, 2006. 92–99 p. Citado na página 28.

LEE, S.; KIM, J. *Warningbird: Detecting suspicious urls in twitter stream*. [S.l.]: NDSS, 2012. Citado na página 29.

LIN, J. bin. *Anomaly Based Malicious URL Detection in Instant Messaging*. [S.l.]: Department of Computer Science and Engineering, National Sun Yat-Sen University, 2008. Citado na página 44.

Ma et al, J. *Beyond blacklists: learning to detect malicious web sites from suspicious urls*. [S.l.]: 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009. 1245–1254 p. Citado 5 vezes nas páginas 26, 28, 29, 38 e 45.

Ma et al, J. *Identifying suspicious urls: an application of large-scale online learnin*. [S.l.]: 26th Annual International Conference on Machine Learning, 2009. 681–688 p. Citado 3 vezes nas páginas 28, 29 e 45.

MA et al, J. *Learning to Detect Malicious URLs*. [S.l.]: Transactions on Intelligent Systems and Technology, 2011. 1245-1254 p. Citado na página 45.

Maggi et al, F. *Two years of short urls internet measurement: security threats and countermeasures*. [S.l.]: 22nd International World Wide Web Conferences Steering Committee, 2013. 861–872 p. Citado na página 29.

MCGRATH, D. K.; GUPTA, M. *Behind phishing: An examination of phisher modi operandi*. [S.l.]: LEET, 2008. 4 p. Citado 3 vezes nas páginas 16, 29 e 43.

Mohammad et al. *Intelligent rule-based phishing websites classification*. [S.l.]: Information Security, IET 8.3, 2014. 153-160 p. Citado na página 43.

Mohri et al. *Foundations of machine learning*. [S.l.]: MIT press, 2012. Citado na página 33.

OLIVO, C. *Avaliação de características para detecção de phishing de email*. 2010. Citado na página 15.

OLIVO, C. K. *Avaliação de Características Para Detecção de Phishing de E-mail*. PUCPR, 2010. Disponível em: <http://www.inf.ufpr.br/lesoliveira/download/CleberOlivoMSC.pdf>. Citado na página 37.

OLLMANN, G. *The Phishing Guide: Understanding & Preventing Phishing Attacks*. IBM Int. Sec. Sys, 2007. Disponível em: <http://www-935.ibm.com/services/us/iss/pdf/phishing-guide-wp.pdf>. Citado na página 21.

OpenDNS. *Phishtank*. 2016. Último acesso em 02 de Junho de 2017. Disponível em: <https://www.phishtank.com>. Citado na página 44.

PATIL, D. R.; PATIL, J. *Survey on malicious web pages detection techniques*. [S.l.]: International Journal of u-and e-Service, Science and Technology, 2015. 195-206 p. Citado na página 16.

Prakash et al, P. *Phishnet: predictive blacklisting to detect phishing attacks*. [S.l.]: INFOCOM. IEEE, 2010. 1–5 p. Citado na página 28.

RESENDE, S. *Sistemas Inteligentes: Fundamentos e Aplicações*. [S.l.]: Manolé, 2003. Citado na página 31.

RUSSEL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach* . [S.l.]: Prentice-Hall, 2002. Citado na página 31.

Sami Tolvanen and Timo Ala-Kleemola. *WoT*. 2007. Disponível em: <https://www.mywot.com>. Citado na página 44.

Seifert et al, C. *Identification of malicious web pages with static heuristics*. [S.l.]: Telecommunication Networks and Applications Conference, IEEE, 2008. 91–96 p. Citado na página 25.

Sheng et al, S. *Anti-phishing phil: The design and evaluation of a game that teaches people not to fall for phish*. 2007. 88-99 p. Citado na página 37.

Sheng et al, S. *An empirical analysis of phishing blacklists*. [S.l.]: Sixth Conference on Email and Anti-Spam (CEAS), 2009. Citado 2 vezes nas páginas 25 e 39.

Sinha et al, S. *Shades of grey: On the effec-tiveness of reputation-based blacklists*. [S.l.]: 3rd International Conference on IEEE, 2008. 57–64 p. Citado 2 vezes nas páginas 25 e 27.

Stringhini et al, G. *Shady paths: Leveraging surfing crowds to detect malicious web pages*. [S.l.]: ACM SIGSAC conference on Computer & communications security, 2013. 133–144 p. Citado na página 30.

Thomas et al, K. *Design and evaluation of a real-time url spam filtering service*. [S.l.]: Security and Privacy (SP), 2011. 447–462 p. Citado na página 30.

Wang et al, D. *Click traffic analysis of short url spam on twitter*. [S.l.]: 9th Intl Conference on Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom). IEEE, 2013. 250-259 p. Citado na página 30.

Y. Zhang and J. Hong and L. Cranor. *CANTINA: A Content-Based Approach to Detecting Phishing Web Sites*. 2007. Citado na página 38.

Yang et al. *A re-examination of text categorization methods*. [S.l.]: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999. 42-49 p. Citado 2 vezes nas páginas 13 e 34.

Zander et al. *The relationship between precision-recall and roc curves*. [S.l.]: Proceedings of the 23rd international conference on Machine learning, 2006. 233–240 p. Citado 3 vezes nas páginas 13, 33 e 34.