Adoption of Feature Selection as Anti-phishing Mechanism: Applicability and Impacts

Mateus L. S. D. Barros[1], **Carlo M. R. Silva**[2], **Péricles B. C. de Miranda**[1]

[1]Departamento de Computação – Universidade Federal Rural de Pernambuco (UFRPE)
Recife – PE – Brazil

[2]Universidade de Pernambuco
Garanhuns – PE – Brazil

{pericles.miranda}@ufrpe.br, {revoredo}@gmail.com

Abstract. *Phishing websites are fake page that deceive victims by passing on legitimate from banks or companies to obtain personal information without their consent. Although learning algorithms have been widely used for phishing detection, there is no consensus as to what attributes are relevant to a better description of a malicious page. This article presents an experimental study that investigates and analyzes the degree of relevance of attributes in different phishing databases. The results showed that a suitable methodology for the selection of attributes is able to reduce the computational cost of the classification process, and still to reach satisfactory results of accuracy and F1 Score.*

Resumo. *Phishing websites são páginas falsas que enganam as vítimas, passando-se por sites legítimos de bancos ou empresas para obterem informações pessoais sem o consentimento delas. Embora algoritmos de aprendizagem tenham sido largamente utilizados para a detecção de phishing, não existe um consenso sobre que atributos são relevantes para uma melhor descrição de uma página maliciosa. Este artigo apresenta um estudo experimental que investiga e analisa o grau de relevância de atributos em diferentes bases de dados de phishing. Os resultados mostraram que uma metodologia adequada para a seleção de atributos é capaz de reduzir o custo computacional do processo de classificação, e ainda alcançar resultados satisfatórios de acurácia e F1 Score.*

1. Introdução

No mundo moderno, a segurança de informações privadas é um dos aspectos mais importantes pelo qual as pessoas prezam. Sendo assim, criminosos praticam ataques de engenharias sociais utilizando a interação humana para a manipulação das vítimas, levando-as à exposição de seus dados confidenciais e adquirindo-os para uso impróprio. Phishing é um desses ataques de engenharias sociais, que visa a captura das informações através do disfarce, passando-se por entidades confiáveis. Dentre as plataformas mais comuns de ataques por phishing, as comunicações eletrônicas, tais como e-mails ou mensagens de texto, se provaram as mais populares, fazendo crescer a eficiência do golpe, e aumentando seu alcance de vítimas [Jagatic et al. 2007]. Os criminosos, na maioria das vezes, se passam por grandes empresas, bancos, organizações ou próprios sites de compras online. Os

e-mails ou mensagens contêm links a inicialização de malwares ou para sites maliciosos que se passam por legítimos, visando a captura dos dados pessoais.

Existe uma série de medidas anti-phishing que não envolvem diretamente a computação. Em [Mohammad et al. 2015] é mostrado que países como EUA, Inglaterra, Canadá e Austrália aprovaram leis que caracterizavam phishing como um crime cibernético, com sentenças de prisão para os criminosos que estivessem atuando na área. Porém, o tempo de vida de sites maliciosos são curtos, impossibilitando a busca por hackers em fontes ainda ativas. Outro método também aplicado foi a educação para a população sobre prevenção de ataques por phishing. Também foi um método não muito eficiente, devido ao fato de ter uma lenta expansão, e por os hackers sempre desenvolverem novos meios de manipulação [Mohammad et al. 2015].

Diante da problemática, pesquisadores passaram a considerar abordagens computacionais, como Aprendizagem de Máquina (AM), para a detecção automática de phishing. AM é uma subárea da ciência da computação que consiste fazer com que sistemas computacionais possam aprender com experiências (dados), e usar este aprendizado para detectar padrões para fins de classificação e ou regressão [Tan et al. 2017]. O AM é apropriado para a detecção de phishings, visto que ele transforma o problema em uma tarefa de classificação. O método reconhece padrões aprendendo com bases de dados de sites maliciosos existentes, podendo ser usado em navegadores de internet para verificar a legitimidade dos sites em tempo real para avisar ao usuário. Essas bases de dados que contém informações de sites maliciosos são compostas por atributos que correspondem a características extraídas do código fonte das páginas. Muitos deles são comuns em páginas maliciosas, o que ajuda o algoritmo a identificar o phishing.

Todavia, a quantidade de atributos por base de dados gera uma quantidade massiva de dados, o que implica em um maior custo computacional, e em alguns casos, dificuldades na classificação causadas pela presença de dados irrelevantes. Sendo assim, este trabalho realiza um estudo experimental que analisa a relevância dos atributos existentes, presentes em repositórios públicos, para a detecção de phishing. O principal objetivo desta pesquisa é recomendar um conjunto de atributos relevantes, e fornecer um panorama quantitativo e qualitativo destes atributos em relação aos que já foram propostos na literatura. Quantitativamente, os resultados alcançados mostraram que o conjunto de atributos selecionados neste trabalho consegue reduzir o custo computacional da tarefa e, ainda assim, alcançar bons valores de acurácia e *F1 Score*. Qualitativamente, identificamos um padrão entre os atributos, nos permitindo categorizá-los em termos de relevância: Muito Alta, Alta, Média e Baixa.

Este artigo está dividido na seguinte forma: a Seção 2 apresenta o referencial teórico e trabalhos relacionados à classificação de phishing; na Seção 3 apresentamos a metodologia adotada para a seleção de atributos; na Seção 4, é apresentada a implementação da metodologia adotada; na Seção 5, encontra-se os experimentos e resultados; e na Seção 6, a conclusão e trabalhos futuros.

2. Detecção de Phishing em Páginas Web

2.1. Referencial Teórico

De acordo com [Fette et al. 2007], alguns métodos foram propostos para tentar deter o phishing. O primeiro deles foi através das toolbars em navegadores. Apesar de apresentar

uma acurácia média de 85%, esse método logo apresentou duas desvantagens principais. A primeira seria a quantidade reduzida de informação contextual fornecida pelos e-mails analisados, impedindo uma análise profunda como um filtro de mensagens. A segunda seria a incapacidade de fornecer uma proteção devidamente segura, pois o usuário ainda poderia facilmente ignorar o aviso sobre o possível phishing.

O segundo método foi através do próprio filtro de e-mail. Apesar de ser um método mais robusto que o anterior, as características analisadas pelo filtro diziam respeito à linguagem e elementos do texto ou no máximo a presença de URLs baseadas em IP ou elemento específico de HTML, o que terminava por gerar uma alta quantidade de falsos negativos.

A medida que novas técnicas para conter os ataques foram surgindo, hackers começaram a desenvolver novos meios de phishing. Em [Banu and Banu 2013] é exemplificado alguns desses tipos. Os mais comuns são os chamados clones phishing, que se passam por sites de corporações reais, pedindo os dados sigilosos das vítimas. Outro tipo similar é o spear phishing, que atua como o modelo passado, porém visando um grupo específico de pessoas, ao invés do público geral. DNS-Based phishing, mais conhecido como pharming, é uma técnica perigosa que faz com que o nome de domínio de um site legítimo seja mapeado no enderço IP de um site malicioso. Já o ataque Man-In-The-Middle ocorre quando o hacker intercepta uma mensagem da empresa original para o usuário e a modifica, inserindo o link para o site ou malware.

Foi devido à diversidade de ataques de phishing que foi considerada a utilização de sistemas inteligentes como AM para a detecção de sites maliciosos [Abdelhamid et al. 2014]. A seguir serão detalhados os principais trabalhos que utilizaram AM para o problema em questão.

2.2. Trabalhos Relacionados

O problema de detecção de páginas maliciosas tem se tornado cada vez mais relevante e estudado. Deste modo, ao longo do tempo, diferentes bases de dados foram criadas com diversos atributos para caracterizar estes tipos de sites. Em [Abdelhamid et al. 2014] são apresentados alguns dos atributos mais usados em bases de dados públicas e que são estudados no presente trabalho. O atributo $having\_IP\_Address$ simboliza o uso de endereço IP no nome de domínio da URL, o que significa que quem criou o site está tentando acessar informações pessoais. Há também os atributos voltados para a construção da URL, como a verificação de @ no corpo ($having\_at\_Symbol$) ou a colocação de prefixos ou sufixos disfarçados entre os símbolos (Preffix_Suffix). Outro exemplo se refere à presença de links que redirecionam à outras páginas de domínio diferente (URL_of_Anchor). Existem uma variedade de atributos que contribuem para a classificação das páginas maliciosas, e estes serão detalhados ao longo do artigo.

As bases de dados disponíveis, que foram construídas ao longo do tempo, exercem um papel fundamental para a aplicação de algoritmos de aprendizagem de máquina, e para o treinamento de modelos de classificação de páginas maliciosas. A seguir, serão apresentados trabalhos que utilizaram AM para o problema o questão.

Em [Mohammad et al. 2013] foi criado um modelo para a predição de phishing baseado em Redes Neurais Artificiais (NN). O resultado do experimento mostrou resultados promissores quanto à acurácia. Em [Mohammad et al. 2014] foi proposto um modelo

inteligente para prever ataques de phishing baseados em NNs, particularmente redes neurais auto-estruturantes. Em [Sumathi and Prakash 2012], foi utilizado um algoritmo de otimização para configurar algoritmos de classificação com os melhores parâmetros. Os resultados obtidos mostraram que o método usado por eles acabou por atingir uma maior acurácia e eficiência em comparação aos métodos clássicos de classificação escolhidos.

Em [Ibrahim and Hadi 2017] é realizada uma comparação entre os classificadores Prism, Multi-Layer Perceptron, Naive Bayes, KStar e Random Forest. Foi utilizado uma base de dados com 31 atributos, e o experimento de validação cruzada 10-fold para obter a acurácia. Como resultado, o Random Forest obteve a melhor média com 95,2%, e o Prism obtendo a pior com 87,6%. Em [Fadheel et al. 2017] foi realizada tanto uma seleção de atributos quanto uma comparação entre classificadores. Para a seleção, usaram o método Kaiser-Meyer-Olkin (KMO), reduzindo a base de dados de 30 para 19 atributos, com os melhores sendo https_token e mouseover. Os classificadores escolhidos foram Logistic Regression (LR) e Support Vector Machines (SVM). Os resultados indicaram uma acurácia de 89% para LR e 92% para SVM, com a acurácia e *F1 Score* sendo respectivamente 90,8% e 92,8%.

Os trabalhos supracitados buscam em sua maioria avaliar os classificadores, otimizar seus parâmetros ou propor novos métodos de classificação para o problema em questão. Ainda existe uma carência de trabalhos que investigam a relevância dos diversos atributos utilizados para a detecção de phishing. Diante disso, este trabalho se propõe a realizar um estudo experimental sobre a relevância dos atributos existentes para a classificação de phishing. O objetivo com este estudo é selecionar aqueles atributos que se mostraram relevantes, promovendo um bom desempenho do classificador e reduzindo o custo do experimento. Para isso, foi definida uma metodologia para a seleção de atributos, que será apresentada a seguir.

3. Metodologia

Nos dias atuais, não há um consenso sobre quais os atributos mais adequados para a detecção de phishing [Fadheel et al. 2017]. Este artigo propõe uma análise aprofundada dos principais atributos utilizados na literatura, de modo a recomendar aqueles mais relevantes, a partir de bases de phishings provenientes de repositórios públicos. A principal contribuição deste trabalho é recomendar atributos que sejam representativos, alcançando um menor custo computacional, mas de forma a não afetar o desempenho do classificador. Para a realização de tal análise, definimos uma metodologia, cujas etapas são ilustradas na Figura 1.

O *pipeline* recebe como entrada um conjunto de bases de dados de sites maliciosos, que podem possuir alguns atributos em comum ou não. Estas bases são fornecidas para o módulo de *Seleção de atributos*, responsável pela identificação de atributos, de cada base de dados, que sejam relevantes e representativos para a detecção de phishing.

Após a seleção dos atributos relevantes de cada base de dados fornecida, a próxima etapa é a *Criação de uma nova base*. Esta nova base é a composição das instâncias das bases participantes, composta por valores que dizem respeito aos atributos que foram selecionados na etapa de *Seleção de atributos*. Uma vez criada a nova base de dados, esta é utilizada no processo de treinamento do(s) modelo(s) de classificação. Além disso, experimentos e testes podem ser configurados para avaliar o modelo, e a contribuição da

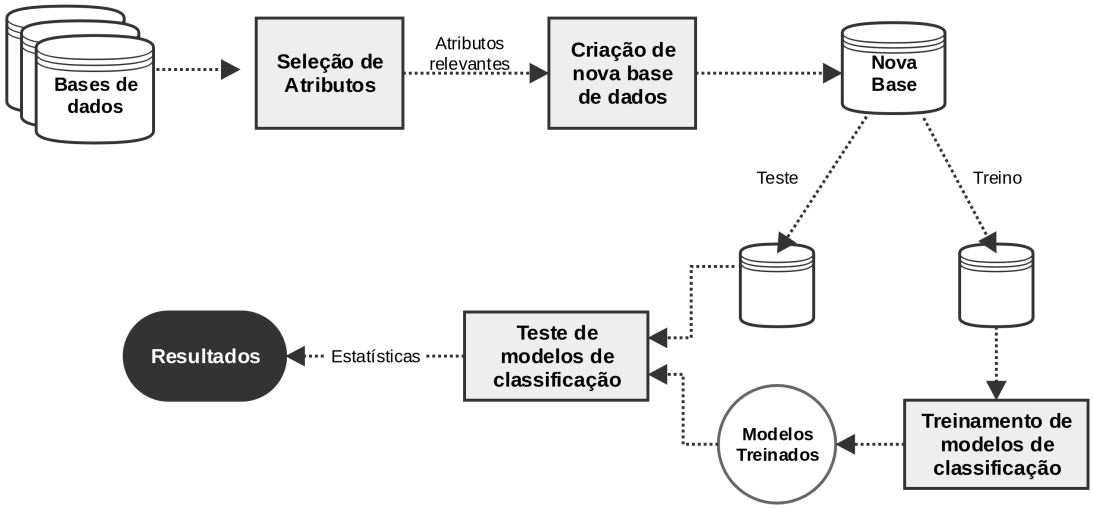

Figura 1. *Pipeline* da metodologia adotada.

nova base de dados no desempenho do mesmo. Mais detalhes de como cada etapa desta metodologia foi implementada, neste trabalho, serão apresentados a seguir.

4. Implementação da Metodologia

4.1. Bases de dados

Neste trabalho foram selecionadas quatro bases de dados de phishing, três contendo 30 atributos e uma contendo 9 atributos. As bases são provenientes de repositórios públicos. As com 30 atributos são: *Phishing website dataset* [Kumar 2018] e *Website Phishing Dataset* [Çelik 2018] do Kaggle e, *Phishing Websites Data Set* [Dua and Graff 2017] do UCI. A base com 9 atributos foi criada no trabalho de [Abdelhamid et al. 2014], e está disponível no UCI. Estas bases foram fornecidas como entrada para o módulo de *Seleção de atributos*. Vale salientar que as 3 bases com 30 atributos possuem os mesmos atributos, mas instâncias diferentes. A base com 9 atributos também possui instâncias diferentes das demais, e todos os seus atributos também estão contidos no conjunto dos 30.

4.2. Seleção de atributos

Nesta etapa é realizada a verificação do grau de contribuição de cada atributo em cada uma das base de dados, e a remoção dos mais irrelevantes. Uma seleção de atributos adequada reduz a dimensionalidade do problema, podendo favorecer dois aspectos: o tempo de treinamento do classificador, e a manutenção da representatividade dos dados na classificação [Dash and Liu 1997].

Neste trabalho, o algoritmo *Extremely Randomized Trees* [Geurts et al. 2006] foi utilizado na seleção de atributos. Este algoritmo tem similaridades com o método *Random Forest*, porém, abandona a ideia de *bootstrapping*, selecionando um ponto de corte aleatório da amostra de aprendizado. Essa ideia favorece a obtenção de uma boa acurácia, reduzindo a variância, fazendo desse algoritmo uma boa alternativa para a seleção de atributos. Como resultado, este algoritmo retorna o grau de contribuição de cada atributo, permitindo ao especialista decidir quais deles serão de fato selecionados.

Aplicamos essa seleção nas três bases de dados de 30 atributos. Dos 30 atributos, 9 foram selecionados, e aqueles que possuíram contribuição abaixo, foram descartados. Como esperado, a seleção de atributos aplicada às três bases retornou o ranking de atributos relevantes idêntico, uma vez que estas bases possuem os mesmos atributos, mesmo com instâncias diferentes. Na Figura 2 são apresentados os 15 atributos mais relevantes, sendo apenas os 9 primeiros, selecionados.

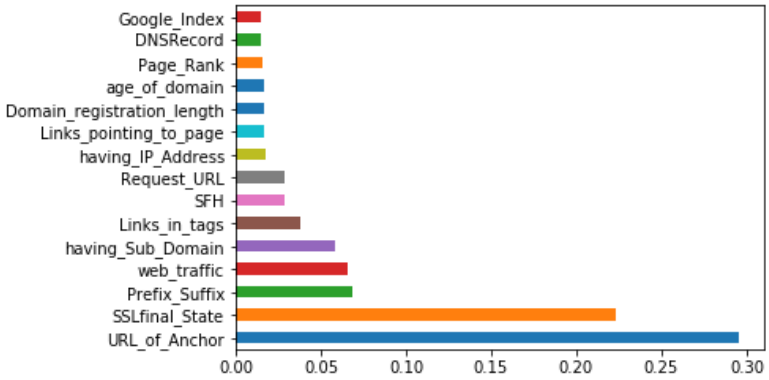

Figura 2. Os 15 atributos mais relevantes das bases selecionadas com 30 atributos.

O mesmo procedimento foi realizado sobre a base com 9 atributos. O resultado se encontra na Figura 3.

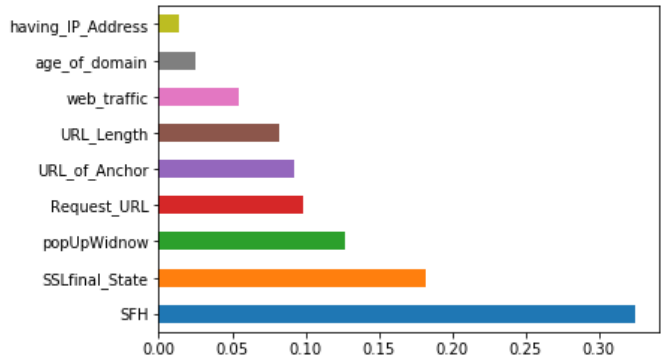

Figura 3. Relevância dos 9 atributos.

Como se pode ver, dentre os 9 atributos selecionados das bases maiores, 6 deles também foram bem ranqueados na base menor. Para fins de geração de uma nova base, realizamos a união entre os atributos das bases, totalizando 12 atributos selecionados.

4.3. Geração de nova base de dados

Tendo em mãos o resultado da etapa anterior, é iniciado o processo para a criação de uma base de dados nova. Para isso, unificamos as instâncias das bases utilizadas considerando apenas os valores dos 12 atributos selecionados, totalizando aproximadamente 6 mil instâncias de sites maliciosos.

A nova base passou a ser constituída pelos 9 atributos das bases maiores, com mais as 3 presentes somente na base menor. Dessa forma, a base resultante é composta

pelos atributos: having_IP_Address, URL_Length, Prefix_Suffix, having_Sub_Domain, SSLfinal_State, Request_URL, URL_of_Anchor, Links_in_tags, SFH, popUpWindow, age_of_domain e web_traffic. Tendo sido finalizado, realizamos testes com ele, que podem ser encontrados na seção 5.

4.4. Algoritmos de classificação

A fim de verificar se a base de dados gerada foi capaz de manter sua representatividade mesmo reduzindo sua dimensionalidade, selecionamos três algoritmos de classificação largamente usados na literatura: Máquinas de Vetor de Suporte (SVM), Árvores de Decisão (DT) e Redes Neurais (NN). Esses algoritmos foram configurados usando seus parâmetros padrão da biblioteca Scikit-Learn [Pedregosa et al. 2011].

5. Experimentos e Resultados

Neste trabalho, foram realizados dois experimentos: 1) Quantitativo - Comparação de resultados obtidos por diferentes classificadores nas diferentes bases de dados envolvidas, considerando-se acurácia e *F1 Score*. Neste experimento comparamos a base de dados gerada a partir da metodologia apresentada (na Seção 3), composta por 12 atributos, três bases com 30 atributos, e 1 base com 9 atributos. 2) Qualitativo - Análise mais aprofundada dos atributos selecionados neste trabalho com aqueles das demais bases de dados. O intuito é identificar padrões entre os atributos e tirar conclusões quanto à indispensabilidade de cada um deles na classificação de de páginas maliciosas. Os experimentos foram executados em um computador Intel Core i7-5500U 2.40GHz, com memória RAM de 8 GB.

5.1. Análise Quantitativa

Os classificadores foram treinados e testados em cada uma das bases utilizando o experimento de validação cruzada 10-fold executado 30 vezes. A partir deste experimento, valores de acurácia e *F1 Score* foram mensurados. A média dos valores de acurácia de cada classificador em cada base de dados pode ser visto na Figura 4.

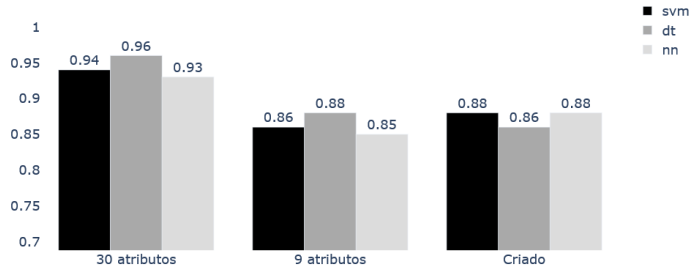

Figura 4. Acurácia média de cada classificador em cada base de dados.

A Figura 4 mostra que a base de dados com 30 atributos atingiu melhores resultados em relação à base com 9 atributos e à proposta (com 12 atributos). Esse resultado é esperado pois a mesma possui quase o triplo de características para descrever o problema. Em contrapartida, esta abordagem torna-se mais custosa experimentalmente, custando 4 vezes mais que a abordagem proposta e 10 vezes mais que a execução na base com 9 atributos. A Figura 6 mostra o custo de execução médio, em segundos, de cada algoritmo em cada base de dados. Além disso, o resultado mostra que a base recomendada (com

12 atributos) foi capaz de superar os valores de acurácia da base com 9 atributos levando em consideração os classificadores SVM e NN. Isso comprova a qualidade dos atributos selecionados.

Embora a acurácia obtida pela proposta tenha alcançado, em média, 8 pontos percentuais a menos quando comparada com a base de 30 atributos, o mesmo não acontece com relação ao *F1 Score*. A Figura 5 apresenta o *F1 Score* médio alcançado por cada cada classificador em cada base de dados.

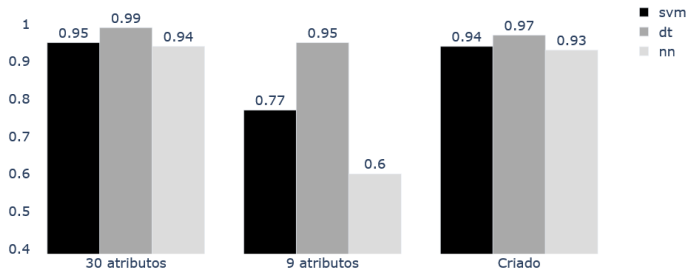

Figura 5. *F1 Score* médio de cada classificador em cada base de dados.

Os resultados alcançados pela proposta são muito próximos daqueles obtidos pela base de dados com 30 atributos, fazendo uso de menos computação. Quando comparada à base com 9 atributos, seus resultados são ainda mais expressivos, superando os resultados em todos os classificadores. Este panorama mostra que os atributos selecionados na proposta foram capazes de reduzir o número de falsos positivos e falsos negativos, resultado importante para o contexto de detecção de sites maliciosos.

Como mencionado na Seção 2.1, o trabalho desenvolvido por [Fadheel et al. 2017] também investigou a seleção de atributos no problema em questão. Comparando com os resultados em [Fadheel et al. 2017], que obteve 93.59% de acurácia e 92.83% de *F1 Score*, a proposta foi vencida na primeira métrica, mas venceu em *F1 Score*. Um dado importante, que vale salientar, é que [Fadheel et al. 2017] utilizou 19 atributos, enquanto a nossa proposta adotou apenas 12.

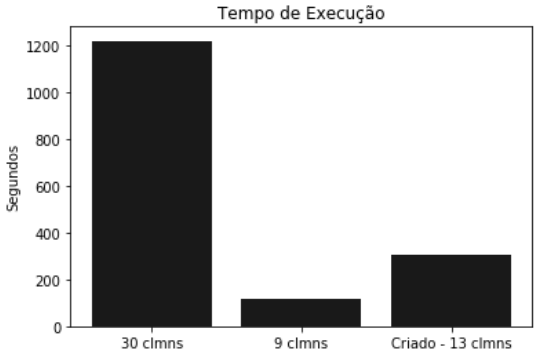

Figura 6. Tempo de execução médio, em segundos, de cada algoritmo em cada base de dados

Visando complementar o experimento quantitativo, aplicamos um algoritmo de redução de dimensionalidade nas 3 bases de dados com 30 atributos, para fins de comparação com a proposta. Algoritmos de redução de dimensionalidade realizam

transformações lineares para delimitar uma quantidade menor de atributos, ignorando os considerados irrelevantes e favorecendo a classificação [Van Der Maaten et al. 2009]. Diante disso, é fundamental comparar a proposta com este tipo de abordagem. O algoritmo de redução escolhido foi o *Principal Component Analysis* (PCA) [Jolliffe 2011]. Vale salientar que a redução de dimensionalidade é uma prática muito comum em trabalhos científicos, visando reduzir a complexidade da base de dados e consequentemente o seu custo. O objetivo desta comparação é verificar se esta redução, no contexto do atual problema, supera os resultados obtidos pela atributos recomendados pela proposta.

Ao aplicar o PCA nas 3 maiores bases, por terem todas os mesmos 30 atributos, o resultado gerado foi o mesmo. Testamos a redução de dimensionalidade considerando três quantidades de componentes diferentes: 15, 9 e 5 atributos, objetivando investigar o impacto desta redução na acurácia e *F1 Score*. As Figuras 7 e 8 apresentam a acurácia e *F1 Score* médios obtidos por cada classificador no problema com 30 atributos reduzido em 15, 9 e 5 atributos através do PCA, respectivamente. Nestas mesmas figuras, foi adicionado o resultado alcançado pela proposta (com 12 atributos), para fins de comparação.

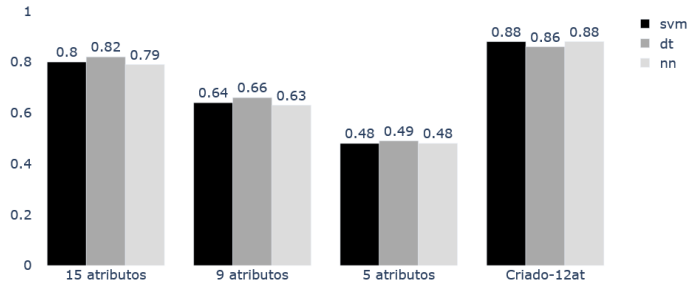

Figura 7. Acurácia média obtida por cada classificador nos diferentes níveis de redução pelo PCA.

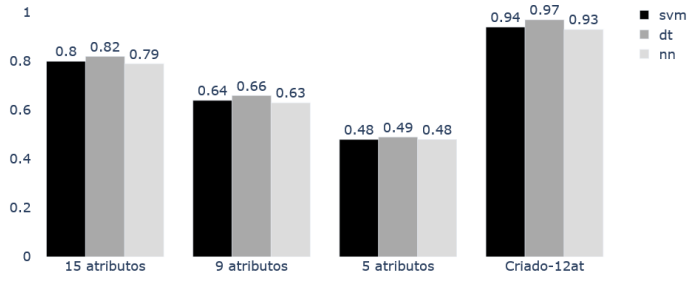

Figura 8. *F1 Score* médio obtido por cada classificador nos diferentes níveis de redução pelo PCA.

Como é possível observar na Figura 7, no cenário em que o PCA reduz a base de 30 atributos para 15, os classificadores SVM, DT e NN alcançaram uma acurácia média de 80%, 82% e 79% respectivamente. Esses valores são inferiores aos obtidos pela proposta com 12 atributos, ou seja, com menos atributos a proposta foi capaz de prover mais informação representativa para os classificadores que a base de dados reduzida com 15 atributos. A superioridade dos resultados da proposta aumenta a medida que a redução de dimensionalidade torna-se maior. Os resultados do PCA com 9 atributos giram em torno de 64%, enquanto que com 5 atributos os resultados atingem valores em torno de 48%. Todos estes resultados estão abaixo dos alcançados pela proposta, independente

do classificador. A comparação entre as reduções de dimensionalidade e a proposta com relação ao *F1 Score* segue a mesma narrativa. Os resultados da proposta com 12 atributos giram em torno de 94%, enquanto que o melhor resultado alcançado pela redução é de 80%.

5.2. Análise Qualitativa

Além da análise quantitativa apresentada previamente, a seguir faremos uma discussão qualitativa dos atributos envolvidos no trabalho.

A fim de ranquear os atributos da nossa base por contribuição, aplicamos o algoritmo *Extra Trees Classifier*, assim como nas bases anteriores (apresentado na Seção 4). Como pode ser visto na Figura 9, dentre os atributos presentes na base criada que vieram do resultado das bases com 30 atributos, Request_URL e SFH tinham sido os menos relevantes. Já no teste com a base de 9 atributos, ambos ficaram entre os mais significativos, com SFH obtendo a maior nota. Na nova base, os dois atributos se estabeleceram em uma colocação intermediária. Um grau de contribuição que não mudou muito foi em relação aos atributos popUpWindow e URL_Length, que ficaram posições intermedárias e/ou derradeiras em todos os testes com as diferentes bases. Sendo assim, se provaram pouco relevantes para a classificação final. URL_of_Anchor, e principalmente SSLfinal_State, por outro lado, ficaram entre os primeiros em todos os testes, mostrando-se indispensáveis em qualquer base de dados para obter uma boa classificação. Prefix_Suffix, having_Sub_Domain e web_traffic se mantiveram em boas colocações ao final.

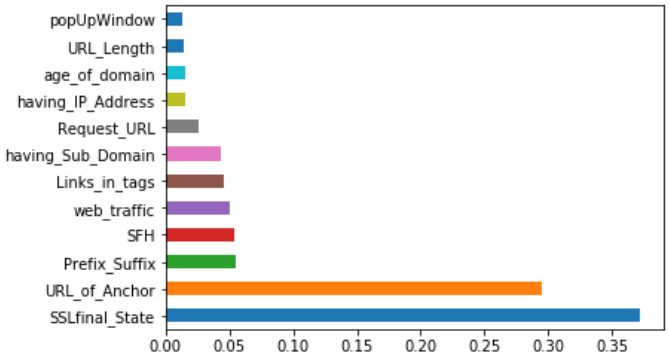

Figura 9. Atributos mais relevantes na base criada.

Isso nos permitiu classificar os atributos em 4 categorias quanto à sua contribuição na classificação de páginas maliciosas: Muito Alta, Alta, Média e Baixa. As duas primeiras categorias são compostas pelos atributos considerados essenciais para qualquer base de dados que queira obter bons resultados; os de categoria Média são os aconselháveis de estarem presentes; e Baixa contribuição são os que podem ser substituídos por outros. A classificação recomendada neste trabalho pode ser vista, em sua completude, na Tabela 1.

A seleção de atributos realizada neste trabalho se mostrou competitiva frente a bases de dados comumente utilizadas na literatura, trabalhos relacionados e bases de dados reduzidas através do PCA. Diante disto, os atributos recomendados pela proposta tornam-se uma alternativa para a classificação de páginas maliciosas.

Relevância	Atributos
Muito Alta	SSLfinal_State
	URL_of_anchor
Alta	Preffix_Suffix
	having_Sub_Domain
	web_traffic
	links_in_tags
	SFH
Média	Request_URL
	age_of_domain
	having_IP_Address
Baixa	popUpWindow
	URL_Length

Tabela 1. Classificação dos atributos finais por relevância.

6. Conclusão e Trabalhos Futuros

A seleção de atributos se tornou uma tarefa indispensável na ciência dos dados e aprendizado de máquina. Neste trabalho realizamos um estudo experimental que investigou o grau de relevância de atributos em diferentes bases de dados de phishing. Quantitativamente, os resultados mostraram que uma seleção de atributos adequada é capaz de reduzir o custo computacional do processo de classificação, e ainda alcançar resultados competitivos de acurácia e *F1 Score* frente a outras bases comumente usadas na literatura, trabalhos relacionados, e bases reduzidas através do PCA. Qualitativamente, identificamos um padrão entre os atributos, nos permitindo categorizá-los em termos de relevância. Os destaques positivos vão para os atributos SSLfinal_State e URL_of_Anchor, que se mostraram os melhores. Enquanto os menos relevantes foram URL_Length e popUpWindow.

Referências

Abdelhamid, N., Ayesh, A., and Thabtah, F. (2014). Phishing detection based associative classification data mining. *Expert Systems with Applications*, 41(13):5948–5959.

Banu, M. N. and Banu, S. M. (2013). A comprehensive study of phishing attacks. *International Journal of Computer Science and Information Technologies*, 4(6):783–786.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, 1(1-4):131–156.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Fadheel, W., Abusharkh, M., and Abdel-Qader, I. (2017). On feature selection for the prediction of phishing websites. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pages 871–876. IEEE.

Fette, I., Sadeh, N., and Tomasic, A. (2007). Learning to detect phishing emails. In *Proceedings of the 16th international conference on World Wide Web*, pages 649–656. ACM.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

Ibrahim, D. R. and Hadi, A. H. (2017). Phishing websites prediction using classification techniques. In *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, pages 133–137. IEEE.

Jagatic, T. N., Johnson, N. A., Jakobsson, M., and Menczer, F. (2007). Social phishing. *Communications of the ACM*, 50(10):94–100.

Jolliffe, I. (2011). *Principal component analysis*. Springer.

Kumar, A. (2018). Phishing website dataset.

Mohammad, R., McCluskey, T., and Thabtah, F. A. (2013). Predicting phishing websites using neural network trained with back-propagation. In *Predicting phishing websites using neural network trained with back-propagation*. World Congress in Computer Science, Computer Engineering, and Applied Computing.

Mohammad, R. M., Thabtah, F., and McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications*, 25(2):443–458.

Mohammad, R. M., Thabtah, F., and McCluskey, L. (2015). Tutorial and critical analysis of phishing websites methods. *Computer Science Review*, 17:1–24.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Sumathi, R. and Prakash, M. R. V. (2012). Prediction of phishing websites using optimization techniques. *International Journal of Modern Engineering Research (IJMER)*, pages 341–348.

Tan, C. L., Chiew, K. L., et al. (2017). Phishing webpage detection using weighted url tokens for identity keywords retrieval. In *9th International Conference on Robotic, Vision, Signal Processing and Power Applications*, pages 133–139. Springer.

Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13.

Çelik, T. (2018). Website phishing dataset.