Detecção de Sites Phishing com Machine Learning:

Uma Abordagem Baseada em Análise de Dados de Sites Autênticos e Maliciosos

Eric Akio Uchiyamada[1], Lucas Goulart de Farias Meres[2], Oliver Kieran Galvão McCormack[3], Pedro Loureiro Morone Branco Volpe[4], Renan Tagliaferro[5]

[1]Ciência da Computação - Faculdade de Computação e Informática - Universidade-Presbiteriana Mackenzie- São Paulo -- SP -- Brasil

{10395287, 10395777, 10395672, 10395922, 10395211}@mackenzista.com.br

Abstract. The proliferation of phishing websites poses a growing threat to online security, creating a need for automated and effective detection methods. This study explores the use of machine learning algorithms to distinguish between phishing and legitimate websites, employing a robust database containing examples of both. Key attributes were extracted from the pages, including URL patterns, which served as inputs for predictive modeling. Various classification algorithms were evaluated based on their accuracy, false positive rate, and generalization ability. The findings indicate that the model with the highest accuracy is not always the best choice, contributing to the development of more reliable and scalable security systems.

Resumo. A proliferação de sites phishing representa uma crescente ameaça à segurança online, gerando a necessidade de métodos de detecção automatizados e eficazes. Este estudo explora o uso de algoritmos de machine learning para distinguir entre sites phishing e sites legítimos, utilizando uma base de dados robusta contendo exemplos de ambos. Foram extraídos atributos característicos das páginas e incluindo padrões de URL, que serviram como entradas para a modelagem preditiva. Diversos algoritmos de classificação foram avaliados quanto à sua precisão, taxa de falsos positivos e capacidade de generalização. Os resultados indicam que nem sempre o modelo com maior acuracidade é o melhor a ser escolhido, contribuindo para o desenvolvimento de sistemas de segurança mais confiáveis e escaláveis.

---

1. Introdução

A evolução da Internet e das transações digitais trouxe consigo desafios de segurança, entre os quais o phishing se destaca como um dos ataques mais frequentes e prejudiciais. Sites phishing são desenhados para se assemelhar a páginas legítimas e enganam usuários para obter informações confidenciais, como senhas e dados bancários. Estima-se que as perdas

anuais com fraudes de phishing ultrapassem bilhões de dólares em todo o mundo, com um impacto negativo significativo tanto para usuários quanto para empresas [Chen et al., 2020].

A detecção de phishing utilizando machine learning tornou-se uma abordagem promissora devido à capacidade desses algoritmos de identificar padrões complexos em grandes conjuntos de dados. Machine learning possibilita a criação de sistemas de detecção mais robustos e dinâmicos, capazes de se adaptar a novos ataques com alta precisão. Este estudo examina diferentes técnicas de machine learning para detecção de phishing, analisando a eficácia de algoritmos populares e explorando as limitações de cada abordagem.

2. Metodologia

Para a condução deste estudo, foram seguidos diversos passos, detalhados abaixo:

2.1 Coleta e Pré-Processamento de Dados

A base de dados utilizada neste estudo inclui amostras de sites phishing e legítimos, obtidas de fontes públicas e confiáveis, como o Zenodo. A extração de características dos sites é um aspecto crucial na detecção de phishing, pois certos padrões são indicadores fortes de práticas fraudulentas. Originalmente, os dados estavam divididos em dois conjuntos separados: um com 5.000 casos de phishing e outro com 5.000 casos de sites legítimos.

Para simular um cenário mais realista — em que a ocorrência de phishing representa cerca de 10%, conforme sugerido por estudos da IBM (IBM, 2023) e relatórios da Verizon Data Breach Investigations Report (DBIR) (Verizon, 2023) —, criamos um novo conjunto de dados combinando os sites legítimos e os de phishing. Este novo dataset contém 5.000 casos de sites legítimos e 578 casos de phishing, totalizando 5.780 amostras. Além disso, para avaliar os modelos desenvolvidos, criamos um dataset adicional de teste, com 6.000 amostras, incluindo 1.000 casos de phishing.

Dessa forma nosso dataset final havia as seguintes colunas com possíveis variáveis preditoras:

- _id, assets_downloaded, brands, domain, features.css, features.html, features.text, folder_path, language, protocol, remote_ip_address, remote_ip_asn, remote_ip_country, remote_ip_domain, remote_ip_isp, remote_ip_isp_org, scan_date, security_issuer, security_protocol, security_state, security_valid_from, security_valid_to, url, whois_domain_age, whois_raw_text, whois_registrar, whois_registrar_url, whois_registry_created_at, whois_registry_expired_at, whois_registry_updated_at, is_phishing

Como o dataset continha várias colunas com variáveis não preditoras e com baixo ganho de informação, optamos por selecionar apenas as colunas mais significativas. Para isso, começamos identificando e removendo variáveis não preditoras — aquelas que não contribuem para prever se um site é phishing ou não, pois não possuem relação direta com a variável-alvo. Exemplos dessas variáveis são "_id", "folder_path", "whois_raw_text" e "whois_registrar", que armazenam informações como identificadores ou o caminho do site na internet. Esses valores são úteis para consultas internas, mas irrelevantes para a análise preditiva.

Após esse filtro inicial, restamos com variáveis mais relevantes para o problema. Para refinar ainda mais, aplicamos a função *mutual_info_classif()*, que calcula o ganho de informação de cada variável. Isso nos permitiu identificar as colunas que realmente contribuem para prever se um site é phishing ou legítimo, conforme mostrado na tabela a seguir.

	Feature	Mutual Information
2	brands	0.319604
4	features.html	0.140281
15	whois_domain_age	0.098412
7	remote_ip_address	0.087660
3	domain	0.068245
14	url	0.064931
1	assets_downloaded	0.064864
9	remote_ip_domain	0.059550
16	whois_registrar_url	0.056360
10	remote_ip_isp	0.053418
11	security_issuer	0.045857
8	remote_ip_country	0.009940
5	language	0.009669
12	security_protocol	0.004609
13	security_state	0.000000

6 protocol 0.000000

Com isso, decidimos remover as colunas com ganho de informação inferior a 0,05. Esse critério garante que nosso dataset final contenha apenas variáveis preditoras relevantes para o modelo, aumentando a eficácia e a precisão da análise.

Dataset após tratamento de dados:

- **"brands":** refere-se à marca da empresa proprietária do site.
- **“features.html”**: atributo relacionado ao formato visual do site, contendo características visuais .html, como "html", "head", "style", "meta", "title", entre outras.
- **“whois_domain_age”:** indica o tempo de atividade do domínio na internet.
- **“remote_ip_address”:** representa o IP remoto remetente ao domínio.
- **“domain”:** o domínio do site em análise.
- **“whois_registrar_url”:** trata-se do domínio responsável pelo registro do site.
- **“url”:** refere-se ao url do site.
- “**assets_downloaded**”: trata-se da quantidade de *assets* baixados no site em questão.

2.2 Seleção de Algoritmos

Foram selecionados algoritmos representativos de diferentes abordagens de aprendizado supervisionado, com foco em modelos que equilibram precisão e interpretabilidade para a detecção de sites de phishing:

- **Regressão Logística**: Este algoritmo linear é amplamente utilizado em tarefas de classificação binária, como a detecção de phishing, devido à sua simplicidade e interpretabilidade. Ele estima a probabilidade de uma instância pertencer a uma classe específica com base em uma combinação linear das variáveis de entrada. Apesar de sua simplicidade, a Regressão Logística é capaz de produzir resultados competitivos em dados que apresentam uma relação linear entre as variáveis independentes e a variável alvo [Hosmer et al., 2013].
- **K-Nearest Neighbors (KNN)**: KNN é um algoritmo baseado em instâncias, que classifica cada amostra com base nas classes das instâncias vizinhas mais próximas no espaço de atributos. Essa abordagem não assume uma forma funcional entre as variáveis, o que pode ser vantajoso em dados com distribuições complexas. Contudo, o desempenho do KNN pode ser impactado por grandes volumes de dados e pela escolha de K, o número de vizinhos a considerar [Cover e Hart, 1967].
- **Random Forest (Floresta Aleatória)**: Este algoritmo é um ensemble de árvores de decisão, onde várias árvores são construídas com diferentes subconjuntos dos dados e dos atributos. Cada árvore contribui com uma “votação” para a classificação final, o

que reduz a variância e aumenta a robustez do modelo. A Random Forest é particularmente útil para dados ruidosos e com variáveis categóricas, características comuns em dados de phishing [Breiman, 2001].

Esses algoritmos foram escolhidos por sua popularidade e desempenho em tarefas de classificação, incluindo a detecção de anomalias e fraudes, sendo particularmente adequados para identificar padrões em dados de phishing.

2.3 Validação e Avaliação

Para garantir a generalização dos modelos, utilizou-se a técnica de validação cruzada em k-partições. A acurácia foi o principal critério de avaliação dos modelos. Depois de realizar o treinamento dos modelos, eles foram testados utilizando o dataset de teste para garantir que os modelos foram treinados corretamente.

Após treinarmos os modelos com o conjunto de dados de treino, obtivemos uma acurácia elevada para cada modelo. Para confirmar que esses resultados realmente indicavam uma boa capacidade preditiva e não eram apenas um caso de overfitting, avaliamos os modelos em um conjunto de dados de teste um pouco maior. Os resultados foram satisfatórios, indicando que os modelos generalizam bem e mantêm a precisão ao prever novos dados.

3. Resultados e Discussão

Após o treinamento e teste dos modelos, foram obtidos os seguintes resultados:

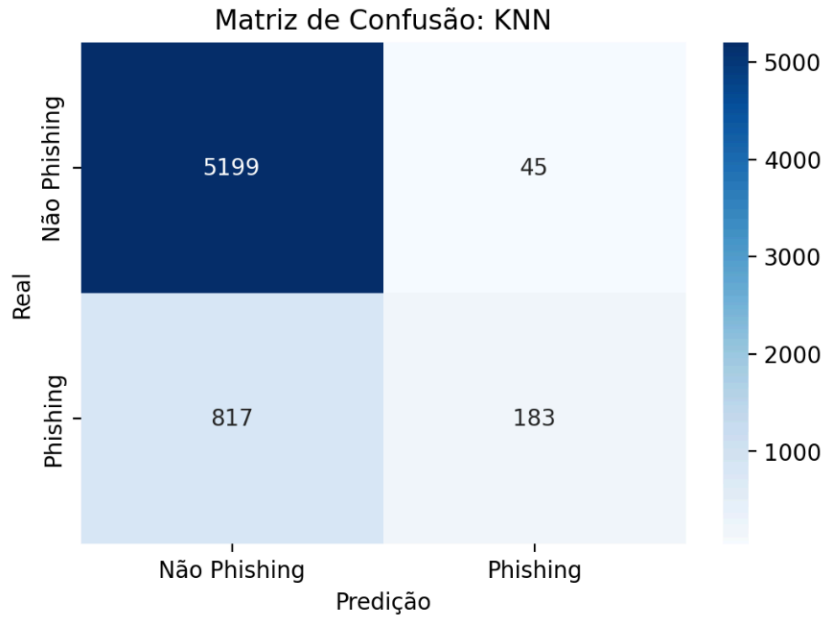

Figura 1. Matriz de confusão do modelo KNN.

Fonte: Elaborado pelo autor.

1. **KNN (K-Nearest Neighbors):** Esta matriz mostra um bom desempenho na classificação de "Não Phishing" (5199 corretas contra 45 incorretas). Porém, o modelo apresentou uma quantidade significativa de falsos negativos (817) ao classificar exemplos de "Phishing" como "Não Phishing". Isso indica que o KNN tem dificuldade em identificar corretamente exemplos de "Phishing" e pode ser considerado conservador, priorizando a classificação como "Não Phishing".

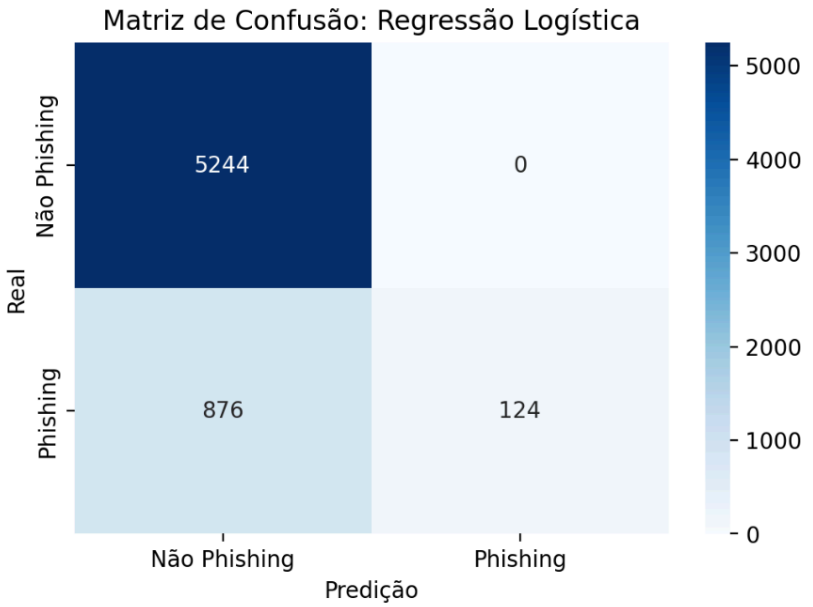

Figura 2. Matriz de confusão do modelo de Regressão Logística.

Fonte: Elaborado pelo autor.

2. **Regressão Logística**: O modelo de Regressão Logística apresenta um alto número de acertos para "Não Phishing" (5244), sem falsos positivos. No entanto, ele também teve muitos falsos negativos (876), indicando que ainda não é eficaz para identificar exemplos de "Phishing". Esse modelo, assim como o KNN, tende a ser conservador, com uma preferência clara para classificar exemplos como "Não Phishing".

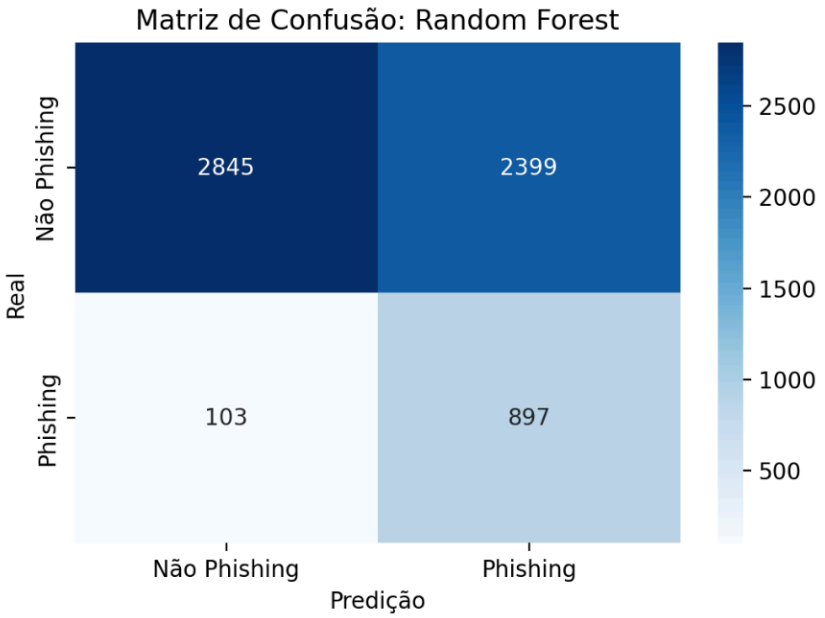

Figura 3. Matriz de confusão do modelo de Random Forest.

Fonte: Elaborado pelo autor.

3. **Random Forest**: A matriz de confusão do modelo Random Forest mostra um melhor equilíbrio entre as classes, com 897 classificações corretas de "Phishing" e uma redução significativa nos falsos negativos (103). No entanto, este modelo apresentou um número maior de falsos positivos (2399), indicando que ele possui uma tendência mais arriscada ao classificar exemplos como "Phishing", o que pode levar a alertas falsos. Em termos gerais, o Random Forest conseguiu capturar melhor os casos de "Phishing" em comparação aos outros modelos.

Na detecção de phishing, falsos positivos (FP) são menos prejudiciais do que falsos negativos (FN), pois os FN deixam sites de phishing passarem despercebidos, expondo usuários a riscos como roubo de dados e fraudes, enquanto os FP causam apenas inconvenientes, bloqueando temporariamente sites legítimos. Estudos apontam que, em segurança cibernética, evitar FN é crucial, uma vez que a exposição a ameaças tem um impacto direto e negativo na segurança do usuário (Sahami et al., 1998; Dhamija et al., 2006). Dessa forma, sistemas de detecção são projetados para minimizar FN, mesmo que isso aumente ligeiramente os FP, priorizando a proteção e a confiança dos usuários em ambientes digitais arriscados (Abdelhamid et al., 2014).

4. Conclusão

Após a análise dos resultados dos modelos desenvolvidos, observou-se que todos apresentaram algumas características indesejáveis. Sendo assim, é preferível selecionar o

modelo que oferece menor risco ao usuário que o utiliza. Dessa forma, conclui-se que, embora os modelos KNN e Regressão Logística tenham apresentado acurácias gerais mais elevadas em comparação ao Random Forest, este último demonstrou uma acurácia superior especificamente em casos de sites de phishing. Além disso, o modelo Random Forest apresentou uma quantidade significativamente menor de falsos negativos em relação aos demais modelos, o que sugere uma maior segurança para um usuário que dependa dessa ferramenta na detecção de sites maliciosos.

Referências

- Chen, W., Liang, Y., & Li, Q. (2020). *Phishing detection using machine learning models: A review*. Journal of Information Security and Applications, 53, 102517.
- Abdelhamid, N., Ayesh, A., & Thabtah, F. (2014). Phishing detection: A recent intelligent machine learning comparison based on models content and features. *Neural Computing and Applications*, 25(3), 473-483.
- Verizon. (2023). *2023 Data Breach Investigations Report*. Verizon Business.
- IBM. (2023). *Cost of a Data Breach Report 2023*. IBM Security.
- COVER, T.; HART, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, v. 13, n. 1, p. 21–27.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.
- HOSMER, D. W.; LEMESHOW, S.; STURDIVANT, R. X. *Applied Logistic Regression*. 3. ed. Hoboken, NJ: John Wiley & Sons, 2013.
- Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop* (pp. 98-105).
- Dhamija, R., Tygar, J. D., & Hearst, M. (2006). Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (pp. 581-590).