Aplicação de Aprendizado de Máquinas para Detecção de URLs Phishing

Applying Machine Learning to Detect Phishing URLs

Noam Resnick[1]
orcid.org/0009-0007-8930-8915

Carmelo Bastos-Filho[1]
orcid.org/0000-0002-0924-5341

[1]Escola Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil. E-mail: carmelo.filho@upe.br

DOI: 10.25286/repa.v9i1.2773

Como citar este artigo pela NBR 6023/2018: Noam Resnick; Carmelo Bastos-Filho. Aplicação de Aprendizado de Máquinas para Detecção de URLs Phishing. Revista de Engenharia e Pesquisa Aplicada, v.9, n. 1, p. 41-49, 2024. DOI: 10.25286/repa.v9i1.2773

RESUMO

Ataques de phishing são um dos ciberataques mais comuns e difíceis de mitigar de forma automatizada. Nos últimos anos, foram propostas uma série de técnicas de detecção e mitigação automatizadas com sucessos variados. Devido ao grande volume de phishing criado diariamente e um tempo de vida médio baixo, é necessária uma técnica de classificação de phishing que consiga atuar de forma rápida e automática. Nesse projeto, propõe-se utilizar técnicas de aprendizado de máquina para realizar essa classificação. Foram coletados mais de 50.000 urls, com mais de 20.000 pertencentes a páginas web que continham um ataque phishing. A partir dessas urls, foi construído um conjunto de dados contendo 15 atributos para o treinamento, validação e testes de modelos de inteligência computacional. Foi realizado um trabalho em três etapas: construção do conjunto de dados, treino de modelos de classificação, e avaliação do desempenho dos modelos treinados. Foram treinados modelos de classificação utilizando os algoritmos Random Forest, XGBoost e Rede Neural Artificial. As métricas utilizadas para avaliar o desempenho dos modelos foram acurácia, precisão e revocação. Os resultados experimentais mostraram o melhor desempenho do modelo de classificação utilizando Floresta Aleatória nas métricas de acurácia, precisão, revocação e F1 Score.

PALAVRAS-CHAVE: Aprendizado de Máquina; Phishing; Segurança da Informação; Classificação; Reconhecimento de Padrões.

ABSTRACT

Phishing attacks are among the most common cyberattacks and are challenging to prevent automatically. Over the last decade many phishing detection and prevention techniques have been proposed with varied degrees of success. Due to the large volume of phishing created daily and a low average life expectancy there is a need for a technique that can reliably and quickly classify phishing urls. In this paper we proposed a phishing attack detection technique based on machine learning. We collected over 50,000 urls with over 20,000 of them belonging to phishing websites. Using these urls we built a dataset containing 14 attributes used for training, validating, and testing of machine learning models. This paper was done in three steps: training classification models, training clustering models, and applying the results of the clustering models to better train the classification models. Three models were trained using the Random Forest, XGBoost and Artificial Neural Network algorithms. The metrics used for the performance evaluation of the trained models were accuracy, precision, and recall. The experimental results show that the best performance can be achieved by a classification model using the XGBoost algorithm.

KEY-WORDS: Machine Learning; Phishing; Information Security; Classification; Pattern Identification.

1 INTRODUÇÃO

O interesse com a cibersegurança é crescente devido ao constante uso e dependência da Internet na sociedade em setores críticos como os bancário e saúde. A segurança digital inclui ações de defesa contraofensivas para obtenção de dados. Existem diversos tipos de ataques digitais que causam prejuízo às vítimas, Biju et al. **[1]** especificaram 10 tipos diferentes, entre eles DDoS, MitM e Phishing.

Phishing é um dos ataques mais comuns visando obter dados privados como credenciais de acesso ou dados pessoais do usuário, esses ataques envolvem sites falsos, ofertas falsas, e-mails de cobrança e mais **[2]**. Os sites de Phishing existentes no mundo inteiro e atingem vários setores do mercado, sendo que suas características e funcionamento mudam dependendo do alvo em questão **[3]**.

Segundo dados levantados pela empresa de cibersegurança ProofPoint **[4]** no Brasil, mais de 70% das empresas sofreram pelo menos um ataque de phishing em 2022, com esse número chegando a 80% mundialmente, e Segundo Kaspersky **[5]** mais da metade dos sites de Phishings criados deixam de existir apenas 94 horas depois da sua criação. Nesta perspectiva, devido ao volume massivo e baixo tempo de vida dos sites de Phishing existe uma necessidade de automatizar a detecção e classificação desses sites maliciosos.

Uma abordagem utilizada para essa automatização é a utilização de algoritmos inteligentes de classificação, conforme nos propõe **[6]**. Este trabalho testou essa abordagem utilizando os algoritmos Floresta Aleatória, XGBoost e Rede Neural Artificial. A partir da leitura de trabalhos anteriores sobre a pesquisa da assertividade de diferentes algoritmos de classificação, foram escolhidos três algoritmos para essa pesquisa. Os resultados obtidos por Zhang e Liu **[7]** mostram que os algoritmos de aumento de gradiente e Floresta Aleatória tiveram o melhor desempenho, enquanto Chang **[8]** demonstrou o desempenho de uma rede neural MLP.

A proposta deste trabalho é apresentar um modelo de classificação inteligente utilizando uma menor quantidade de atributos que exibe melhores resultados.

2 REFERENCIAL TEÓRICO

Sendo a comunicação digital essencial para o funcionamento da sociedade, o seu uso massivo nas mais diversas formas chama a atenção dos atacantes digitais, e os invasores optam por usar estratégias de engenharia social para comprometer as fontes de informação com fins de obter acessos não autorizados.

Ataques de phishing estão entre os tipos que mais circulam nas redes. Segundo dados levantados pela empresa de cibersegurança ProofPoint **[4]** no Brasil mais de 70% das empresas sofreram pelo menos um ataque de phishing em 2022, com esse número chegando a 80% mundialmente.

Os sites de Phishing existentes no mundo inteiro atingem vários setores do mercado, sendo que suas características e funcionamento mudam dependendo do alvo em questão **[3].**

Explorando a literatura publicada nos últimos 10 anos, Basit et al. **[9]** encontram que as abordagens convencionais para detecção de ataques de phishing fornecem baixa precisão e podem reconhecer apenas cerca de 20% dos ataques. Já as abordagens de aprendizado de máquina fornecem bons resultados para a detecção de phishing, mas são demoradas mesmo em conjuntos de dados de pequeno porte e não podem ser dimensionadas. O reconhecimento de phishing por técnicas heurísticas fornece altas taxas de falsos positivos. Basit et al. **[9]** focou em uma ou mais técnicas para melhorar a precisão. Os autores defendem que a precisão pode ser melhorada ainda mais pela redução de recursos e pelo uso de um modelo de conjunto.

Uma análise comparativa detalhada revelou que os métodos de aprendizado de máquina são os métodos mais usados e eficazes para detectar um ataque de phishing. Técnicas com redução de recursos proporcionam melhor desempenho. A classificação pode ser feita por meio de Floresta Aleatória (RF), Rede Neural MLP, Support Vector Machine (SVM), Regressão Logística (LR), C4.5, K Nearest Neigbors (k-NN) e XGBoost (XGB) **[9]**.

Sordo e Zeng **[10]** mostra que existe uma relação direta entre o tamanho do conjunto de dados utilizados para treinar modelos inteligentes de classificação e os seus resultados. Nesse sentido, foi decidido como importante a criação de um conjunto de dados que tivesse maior quantidade de entradas.

Um elemento importante para a criação de um modelo de classificação é a seleção de features do treinamento, uma vez que os resultados dos classificadores de aprendizado de máquina

dependem fortemente da qualidade dos atributos selecionados **[11]**.

É necessário normalizar os dados coletados para utilização nos algoritmos de aprendizado de máquina. Para tanto, utiliza-se neste trabalho um script Python usando a biblioteca Scikit-Learn **[12]**.

Uma análise dos valores coletados para cada atributo oferece informações sobre o impacto de cada um na classificação resultante, visando a exclusão de atributos pouco relevantes.

Para restringir o número de características neste trabalho, propõe-se a utilização de uma estratégia de seleção de atributos a partir da sua correlação com a classe, similarmente ao descrito por Michalak e Kwasnicka **[13]**.

Neste projeto utiliza-se o mapa de calor para a visualização da correlação dos atributos com a classe.

Além da análise da correlação, outra característica estudada foi a distribuição dos valores de cada atributo. Esse estudo permite a exclusão de atributos que tem pouca ou nenhuma variância.

Os diagramas de caixa permitem visualizar a variância para valores contínuos, enquanto para valores categóricos é mais adequado a utilização de gráficos de barras.

Métodos de classificação com aprendizado de máquina implicam na utilização de algoritmos de treinamento. Neste trabalho, selecionou-se os seguintes algoritmos: RF, XGB e Rede Neural MLP.

RF é um método de aprendizado para classificação, regressão e outras tarefas que funciona a partir da construção de uma combinação de preditores e na escolha de uma quantidade aleatória dentre eles durante o treinamento. O resultado de classificação deste algoritmo é o resultado obtido pela maioria dos preditores para uma dada entrada. Para o treino deste algoritmo existem vários hiperparâmetros que devem ser configurados previamente, tais como quantidade de árvores na floresta, quantidade de características que serão usadas para dividir um nó, profundidade máxima das árvores e método de escolha dos atributos. Nesta pesquisa foram selecionados um conjunto de valores diferentes para os hiperparâmetros utilizando o conjunto com melhor desempenho **[14]**.

Uma melhoria para o algoritmo de RF é conhecida como Tree Boosting, que utiliza conceitos de aumento de gradiente. O algoritmo XGB é baseado nessa técnica e tem mostrado resultados melhores do que o RF para cenários similares. Apesar de ser uma técnica parecida com Floresta Aleatória, ambas sendo algoritmos de aprendizado em conjunto, o XGBoost utiliza impulsionamento de gradiente que vai iterativamente adicionando aprendizes fracos ao modelo em uma tentativa de minimizar o erro geral. O hiperparâmetro principal deste algoritmo é o tipo do impulsionador utilizado, no caso deste projeto foi utilizado o Multiclass Log Loss ou entropia cruzada **[15]**.

Outro algoritmo comumente usado para tarefas de classificação é a rede neural MLP. Este algoritmo não tem finalidade específica de classificação como os anteriores, mas apresenta bons resultados para problemas diferentes. Para o cenário de classificação o resultado terá a forma de um neurônio na camada de saída para cada classe definida **[16]**.

Para realizar a comparação dos resultados entre os diferentes algoritmos selecionados, é necessária a escolha das métricas adequadas. Das métricas utilizadas comumente, neste trabalho optou-se por usar as seguintes: acurácia, precisão, revocação e F1 Score. Cada uma dessas métricas procura analisar uma característica diferente dos resultados obtidos, respectivamente são, quantas classificações foram corretas do total, das classificações positivas quantas foram corretas, das classificações que deveriam estar positivas quantas foram classificadas como tal e uma análise dos resultados de precisão e revocação **[17]**.

Com essa fundamentação teórica é possível propor um processo aperfeiçoado para a detecção de ataques de phishing.

3 METODOLOGIA

Para este projeto foi criado um conjunto de dados específico visando atender a proposta de treinar um modelo de classificação binária de phishing com menor quantidade de atributos.

De acordo com a proposta, durante o projeto foram treinados vários modelos de computação inteligente com o objetivo de selecionar um modelo de classificação com melhor desempenho e menor custo.

Foi definido que o conjunto de dados precisaria ter no mínimo 50.000 entradas únicas, com uma distribuição de classes próxima de 50% de entradas classificadas como phishing.

A partir de uma leitura dos trabalhos de Salahdine et al. **[6]** e Mohammad et al. **[18]** foram escolhidos um total de 15 atributos potencialmente importantes para a classificação.

Os atributos selecionados foram divididos em duas categorias:

DOI: 10.25286/repa.v9i1.2773

1. Atributos contidos diretamente na URL:

- **url_size**: Quantidade de caracteres presentes na URL.
- **qty_dots**: Quantidade de pontos presentes na URL.
- **qty_slashes**: Quantidade de barras presentes na URL.
- **domain_is_ip:** Verdadeiro se a URL for um IP.
- **entropy**: Entropia de Shannon calculado a partir da URL.
- **has_cyrillic**: Presença de caracteres cirílicos na URL.
- **has_query**: Presença de parâmetros na URL.

2. Atributos não presentes na URL:

- **has_ssl**: Verdadeiro se foi registrado um certificado SSL para o domínio contido na URL.
- **days_since_domain_reg**: Quantidade de dias desde o registro do domínio contido na URL.
- **days_since_cert_reg**: Quantidade de dias desde a criação do certificado SSL.
- **has_input_tags**: Presença de tags de input (form, input, button) no HTML do site endereçado pela URL.
- **suspicious_links**: Presença de urls contidos no HTML do site endereçado pela URL que não contêm o domínio da URL e não são conteúdos estáticos.
- **qty_redirects**: Quantidade de redirecionamentos realizados pelo site a partir da URL original.
- **detected_language**: Enumeração do idioma principal detectado no HTML do site endereçado pela URL.

Para estabelecer métricas confiáveis, é necessário realizar um pré-processamento à análise exploratória do conjunto de dados, a fim de encontrar relacionamentos estatísticos entre os atributos.

Para os atributos contínuos, foi realizada uma normalização dos valores entre 0 e 1. Com as métricas definidas é possível treinar os modelos de classificação mais eficientes.

Os algoritmos selecionados para este projeto foram Floresta Aleatória, XGBoost e Rede Neural MLP, em função de serem comumente encontradas na literatura especializada para classificação **[7]**. Durante o processo de desenvolvimento deste trabalho foram criados códigos na linguagem de programação Python utilizando de bibliotecas próprias para ciências de dados, tais como, Scikit-Learn, Tensorflow, Pandas e Matplotlib.

O treino de cada modelo foi realizado 30 vezes, sendo uma para cada faixa de atributos definida no final da etapa da análise e cada treino foi repetido 10 vezes mudando os conjuntos de dados de treino e teste para garantir a convergência dos resultados. Para cada rodada de treinamento o conjunto de dados foi dividido, nos algoritmos de ensamble 70% dos dados foram utilizados na fase de treinamento e 30% na fase de teste. Já para a rede neural MLP a divisão foi de 70% para treino, 20% para validação e 10% para teste. O resultado final de cada modelo foi obtido a partir da média das métricas dos 10 treinamentos.

4 RESULTADOS

4.1 CONJUNTO DE DADOS

O primeiro resultado deste projeto foi a criação de um conjunto de dados para treinamento de modelos de aprendizado de máquina voltado para URLs phishing. O conjunto possui um total de 50.679 entradas únicas, onde cada entrada contém 15 atributos além da classe.

Tabela 1- Amostra do Conjunto de Dados
Fonte: Os autores

entrada	url_size	qty_dots	qty_slashes
1	0.024390	0.0909090	0.07142857
2	0.008130	0	0.07142857
3	0.012195	0.0909090	0.07142857
4	0.010840	0.0909090	0.07142857
5	0.032520	0.0909090	0.07142857
6	0.111111	0.2727272	0.07142857
7	0.111111	0.1818181	0.21428571
8	0.124661	0.0909090	0.21428571
9	0.126016	0.1818181	0.21428571
10	0.117886	0.2727272	0.07142857

Tabela 2- Amostra do Conjunto de Dados
Fonte: Os autores

entrada	domain_is_ip	has_query	has_ssl
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	1
7	0	0	1
8	0	0	1
9	0	0	1
10	0	0	1

Tabela 3- Amostra do Conjunto de Dados
Fonte: Os autores

entrada	entropy	has_cyrillic	days_since_domain_reg
1	0.399266	0	0.66344445
2	0.282183	0	0.29916838
3	0.228593	0	0.71817471
4	0.171463	0	0.64347146
5	0.295880	0	0.29561447
6	0.622509	0	0.03610775
7	0.587381	0	0.23711706
8	0.646351	0	0.13227663
9	0.647347	0	0.12630606
10	0.633483	0	0.08891889

Tabela 4- Amostra do Conjunto de Dados
Fonte: Os autores

entrada	days_since_cert_reg	has_input_tags	suspicious_links
1	0.0025380	1	0.0080044
2	0.0025380	0	0.0209770
3	0.0025380	1	0.0322936
4	0.0025380	1	0.0367099
5	0.0025380	1	0.0436102
6	0.2893401	0	0
7	0.0583756	0	0
8	0.5532994	1	0.0013800
9	0.2182741	0	0
10	0.1294416	0	0

Tabela 5- Amostra do Conjunto de Dados
Fonte: Os autores

entrada	detected_language	qty_redirects	is_phish
1	50	0.0080044162	0
2	9	0.0209770908	0
3	50	0.0322936792	0
4	50	0.0367099089	0
5	15	0.0436102677	0
6	0	0	1
7	50	0	1
8	50	0.0013800717	1
9	50	0	1
10	50	0	1

4.2 ANÁLISE EXPLORATÓRIA

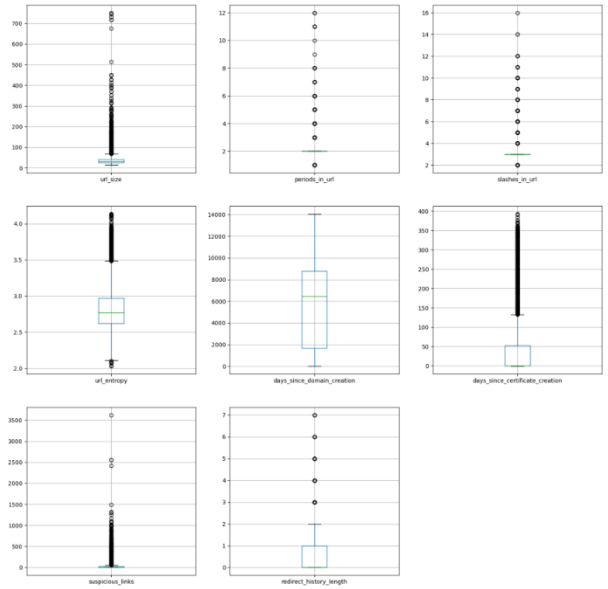
Após o pré-processamento de dados, foi realizada uma análise estatística dos dados coletados.

A análise trabalhou com dois tipos de atributos, aqueles com valores contínuos e os categóricos. Para cada tipo foi utilizada uma técnica de visualização diferente.

Para os atributos do tipo contínuo foram traçados diagramas de caixa, mostrando os valores médios, medianas e outliers.

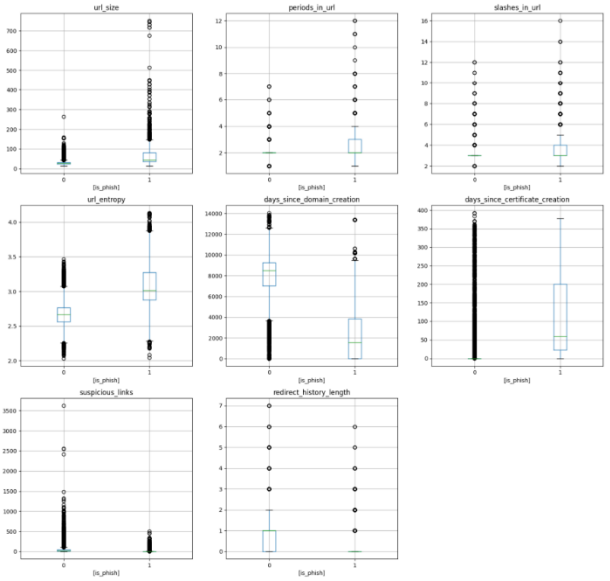
DOI: 10.25286/repa.v9i1.2773

Figura 1 – Diagrama de Caixas de cada Atributo



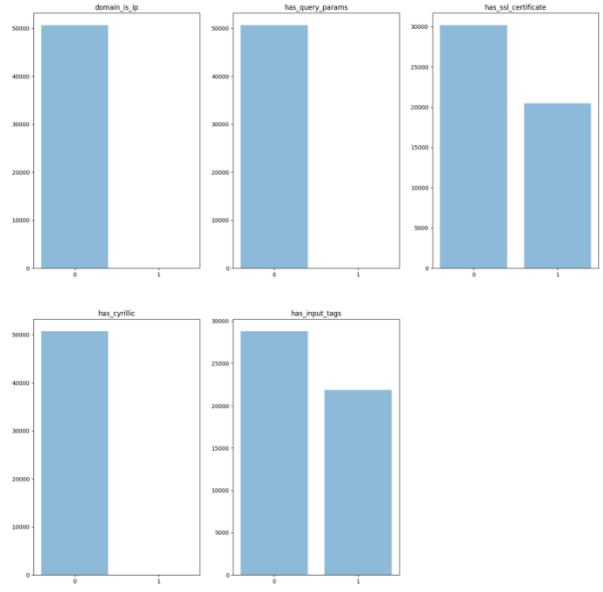
Fonte: Os Autores.

Figura 2 – Diagrama de Caixas de cada Atributo, Separado por Classe



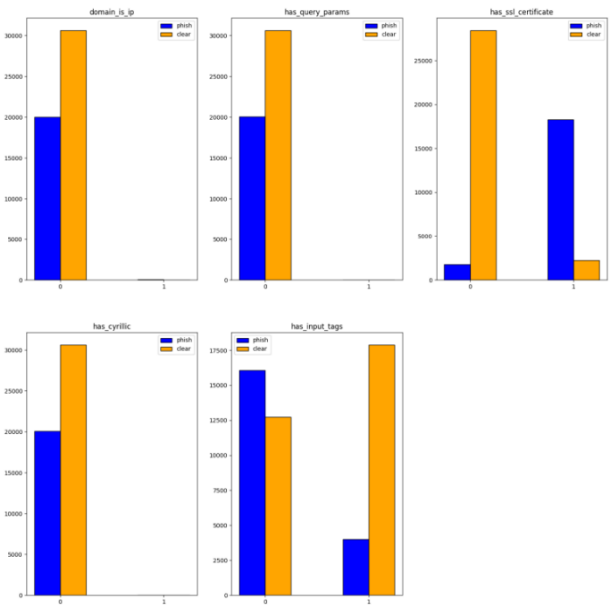
Fonte: Os Autores.

Figura 3 – Gráfico de Barras de cada Atributo



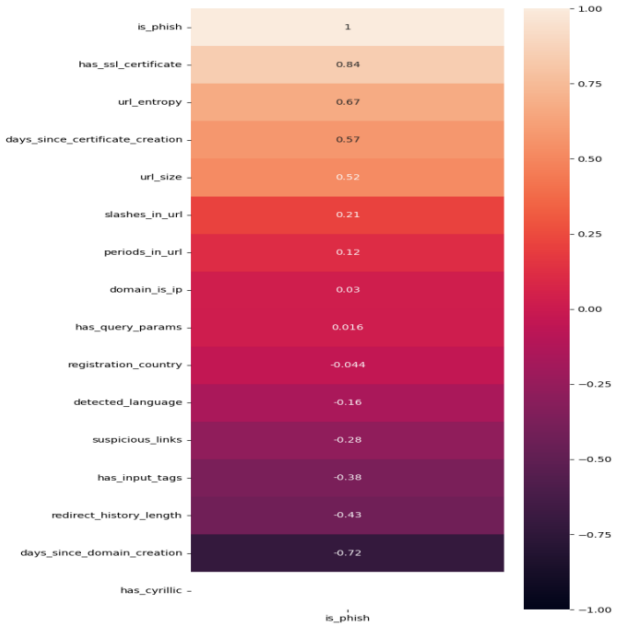
Fonte: Os Autores.

Figura 4 – Gráfico de Barras de cada Atributo, Separado por Classe



Fonte: Os Autores.

Figura 5 – Mapa de Calor da Correlação dos Atributos com a Classe



Fonte: Os Autores.

4.3 MODELOS DE CLASSIFICAÇÃO

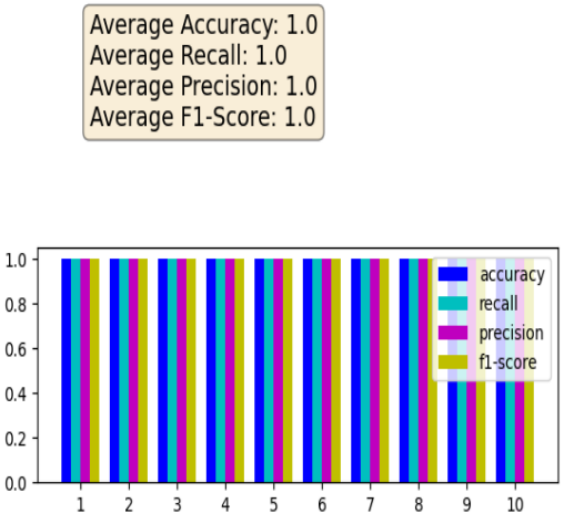
Para a criação dos modelos de classificação, foram escolhidos 3 algoritmos comumente utilizados para esse fim: Floresta Aleatória, XGBoost e Rede Neural MLP. Para cada algoritmo é necessário escolher um conjunto de parâmetros de treinamento, foram selecionados vários conjuntos de parâmetros para cada algoritmo com a finalidade de encontrar os parâmetros com o melhor desempenho.

As métricas utilizadas durante o processo de treinamento dos modelos foram: Acurácia, Precisão, Revocação e F1 Score.

Acurácia mede a assertividade do modelo, levando em consideração todas as classificações realizadas. Precisão leva em consideração apenas as classificações positivas. Revocação responde à pergunta de quantas classificações positivas foram acertadas. F1 Score é um cálculo realizado em cima da Precisão e Revocação.

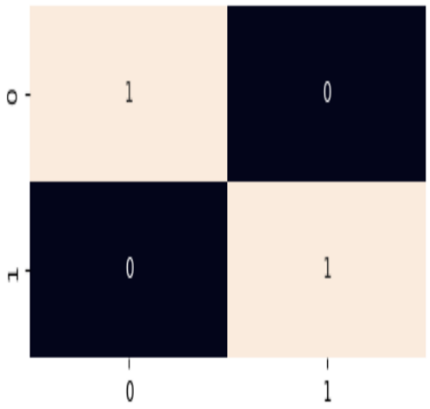
O treino foi realizado utilizando-se da linguagem de programação Python e das bibliotecas conhecidas para aprendizado de máquina, tais como, SkLearn, TensorFlow e Pandas.

Figura 6 – Resultados dos Treinamentos Utilizando Floresta Aleatória



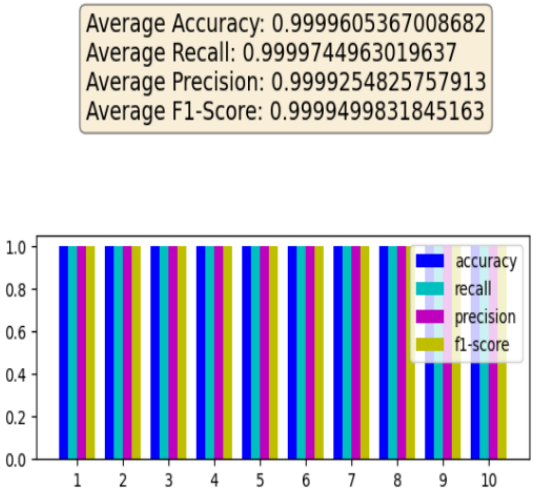
Fonte: Os Autores.

Figura 7 – Matriz de Confusão - Floresta Aleatória



Fonte: Os Autores.

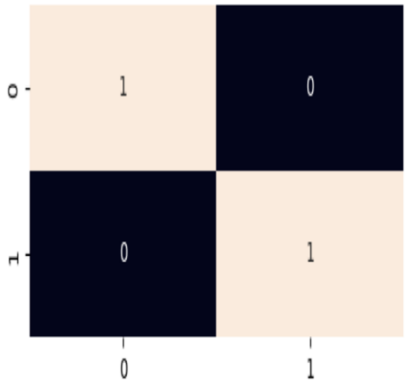
Figura 8 – Resultados dos Treinamentos Utilizando XGBoost



Fonte: Os Autores.

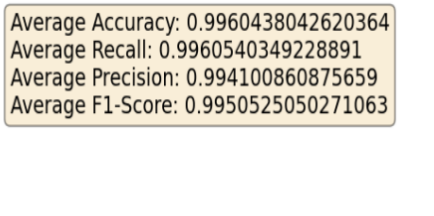
DOI: 10.25286/repa.v9i1.2773

Figura 9 – Matriz de Confusão – XGBoost



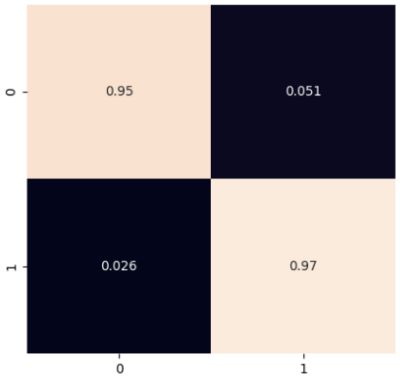
Fonte: Os Autores.

Figura 10 – Resultados dos Treinamentos Utilizando Rede Neural MLP



Fonte: Os Autores.

Figura 11 – Matriz de Confusão - Rede Neural MLP



Fonte: Os Autores.

5 DISCUSSÃO E CONCLUSÕES

Este trabalho aportou interessantes informações para a classificação voltada para a detecção de URLs de phishing.

Durante a análise exploratória, inicialmente ficou evidente que dos 15 atributos coletados, 3 poderiam ser eliminados devido a não apresentarem distribuição de valores significativos (próximo de 100% das entradas no conjunto de dados tinham o mesmo valor).

Analisando o gráfico de correlação dos atributos (Figura 5) notou-se uma dispersão entre os valores, facilitando a escolha dos atributos mais relevantes para o treinamento.

É possível afirmar com base nos resultados obtidos no trabalho que é viável a criação de um modelo de classificação de URLs de phishing com menor quantidade de atributos e maior precisão.

Com as propostas apresentadas neste trabalho, os resultados se mantiveram muito bons mesmo com a diminuição da quantidade de atributos utilizados no treinamento, chegando a 5 atributos com mais de 99% de precisão.

Dos 3 algoritmos utilizados neste trabalho, observou-se que o RF obteve os melhores resultados de acordo com as métricas propostas com 100% de acurácia, precisão, recall e F1 Score.

Encontramos que as principais diferenças entre este projeto e outras propostas encontradas na literatura especializada, a saber, a quantidade de entradas no conjunto de dados, a escolha dos atributos coletados e os algoritmos de classificação utilizados, promovem avanço significativo nos processos de classificação.

Em conclusão, esta investigação mostra-se de grande relevância para a otimização de modelos de classificação de URLs de phishing robustos e eficientes.

REFERÊNCIAS

[1] BIJU, J. M. *et al*. **Cyber attacks and its different types**: International Research Journal of Engineering and Technology 6.3. 2019.

[2] SALAHDINE, F.; KAABOUCH, N. **Social engineering attacks: A survey**: future internet. Signal Processing for Next Generation Wireless Networks. 2019.

[3] ANTI-PHISHING WORKING GROUP. **Phishing Activity Trends Report. 2018**. Disponível em: https://docs.apwg.org/reports/apwg_trends_report_q1_2018.pdf. Acesso em: 12 ago. 2023.

[4] PROOFPOINT. **State of the Phish, 2023: Brasil**. 2023. Disponível em: https://www.proofpoint.com/br/resources/thr

eat-reports/state-of-phish. Acesso em: 12 ago. 2023.

[5] KASPERSKY. **Powerful but short-lived: One third of phishing pages cease to be active after a single day**. 2021. Disponível em: https://usa.kaspersky.com/about/press-releases/2021_powerful-but-short-lived-one-third-of-phishing-pages-cease-to-be-active after-a-single-day. Acesso em: 12 ago. 2023.

[6] SALAHDINE, F.; KAABOUCH, N.; MRABET, Z. E. **Phishing attacks detection a machine learning-based approach**. 2021 IEEE 12th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON). 2022.

[7] ZHANG, C.; LIU, C. **An up-to-date comparison of state-of-the-art classification algorithms**. Expert Systems with Applications 82. 2017.

[8] CHANG, P. C. **Multi-layer perceptron neural network for improving detection performance of malicious phishing urls without affecting other attack types classification**. arXiv preprint arXiv:2203.00774, 2022.

[9] BASIT, A. et al. **A comprehensive survey of ai-enabled phishing attacks detection techniques**. Telecommunication Systems 76. 2020

[10] SORDO, M.; ZENG, Q. **On sample size and classification accuracy: A performance comparison**. International Symposium on Biological and Medical Data Analysis. 2005

[11] ZONGKER, D. E.; JAIN, A. **Algorithms for feature selection: An evaluation**. Proceedings of 13th international conference on pattern recognition. Vol. 2. 1996.

[12] SCIKIT-LEARN. **sklearn.preprocessing.MinMaxScaler. 2023**. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html. Acesso em: 31 ago. 2023

[13] MICHALAK, K.; KWASNICKA, H**. Correlation–based feature selection strategy in classification problems**. International Journal of Applied Mathematics and Computer Science 16.4. 2006.

[14] BREIMAN, L. **Random forests**. Machine learning 45. 2001.

[15] CHEN, T.; GUESTRIN, C. **Xgboost: A scalable tree boosting system**. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

[16] HEPNER, G. F. **Artificial neural network classification using a minimal training set: Comparison to conventional supervised classification**. Photogrammetric Engineering and Remote Sensing 56.4. 1990.

[17] TOWARDS DATA SCIENCE. **Metrics to Evaluate your Machine Learning Algorithm**. 2018. Disponível em: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234. Acesso em: 31 ago. 2023.

[18] MOHAMMAD, R. M.; THABTAH, F.; MCCLUSKEY, L. **Phishing websites features**. School of Computing and Engineering, University of Huddersfield. 2015.

DOI: 10.25286/repa.v9i1.2773