Poster: PhisherCop - An Automated Tool Using ML Classifiers for Phishing Detection

Naheem Noah*, Abebe Tayachew†, Stuart Ryan‡ and Sanchari Das§
Ritchie School of Engineering and Computer Science, University of Denver
Denver, Colorado
Email: *Naheem.Noah@du.edu, †Abebe.Tayachew@du.edu, ‡Stuart.Ryan@du.edu, §Sanchari.Das@du.edu

***Abstract*—Phishing poses a significant security risk to organizations and individuals, leading to the loss of billions of dollars yearly. While risk communication serves as a tool to mitigate phishing attempts, it is imperative to create automated phishing detection tools. Numerous Natural Language Processing (NLP) and Machine Learning (ML) approaches have been deployed to tackle phishing. However, phishing emails and SMS continue to increase exponentially, reiterating the need for more effective approaches. To address this, we have developed an anti-phishing tool called PhisherCop. PhisherCop is built upon a Stochastic Gradient Descent classifier (SGD) and a Support Vector Classifier (SVC) which showed an average accuracy of 96% performing better than six other popular classifiers, including Decision Tree, Logistics Regression, Random Forest, Gradient Boosting Classifier, K-Nearest Neighbors and Multinomial Naive Bayes. With a high level of accuracy, our tool distinguishes between phishing and legitimate content both over emails and text messages based on a user-centered approach.**

***Index Terms*—Phishing, Privacy and Security, User-Centered, Machine Learning, Automated Detection Tools, NLP.**

I. Introduction

Phishing is an act propagated by cybercriminals whereby they send malicious content to individuals to trick them into falling for a scam via several sources, including emails and text messages [2], [4]. However, phishing through SMS or text messages, otherwise called "Smishing" [6] is more concerning because people assume mobile devices are more secure than computers [7]. However, most approaches used by various email or SMS filters today are outdated, the adoption rate is low, and the proliferation of phishing attacks is steadily increasing [1], [9]. While hackers are constantly evolving in their operating techniques, there is a need for more effective and user-centered anti-phishing tools.

We implemented ML and NLP techniques to detect phishing in email and text messages by creating PhisherCop. First, we collected phishing and ham email corpus, performed feature extraction and vectorization, and implemented classification employing eight ML classifiers. They included; the Stochastic Gradient Descent classifier (SGD), Support Vector Classifier (SVC), Decision Tree Classifier (DT), Logistics Regression Classifier (LR), Random Forest Classifier (RF), Gradient Boosting Classifier (GBC), K-Nearest Neighbors Classifier (KNN) and Multinomial Naive Bayes Classifier (MNB). After that, we designed a user-centered web-based phishing detection tool using the classifiers with the highest accuracy score. The tool allows users to paste either their email or SMS content and receive a percentage score indicating its level of legitimacy.

The contributions of our work are;

- We have reviewed existing automated protocols for phishing detection and selected the eight most prominent classifiers that have been previously implemented.
- We implemented the classifiers on our training data and compared the Accuracy score of these classifiers separately on SMS-based phishing content, email address, email body, and subject line to find the best performing classifier.
- Based on our analysis, we designed a user-centered web interface that collects any or all of the content, subject line, and email address of a message. We determined the legitimacy of the content using the best performing classifier in terms of accuracy score.

II. PhisherCop: Implementation Details

Our proposed system PhisherCop is designed to identify emails and SMS as legitimate or phished.

A. Data Collection and Processing

To start implementing the tool, we collected the phishing email corpus by Spam Assassin [1] available at Kaggle [2]. The corpus contains 2551 ham and 501 phishing emails. For the SMS, we collected the Smishing public corpus in Kaggle [3]. The corpus contains 5,574 SMS messages in English tagged as legitimate or phishing. After that, we removed stop words using the set of 127 English stop-words available in the NLTK library, such as "the", "a", "an", "in" [4], emojis, emoticons, punctuation marks, and HTML tags. Next, we implemented stemming using Porter Stemmer, which helps to map related words to the same stem. To transform tokenized words into features, we introduced Term Frequency Inverse Document Frequency (TFIDF) which assigns each word a weight based on its term frequency (TF) and inverse document frequency (IDF) and considers the words with higher weight to be more critical.

[1]https://spamassassin.apache.org/old/publiccorpus/
[2]https://www.kaggle.com/veleon/ham-and-spam-dataset
[3]https://www.kaggle.com/uciml/SMS-spam-collection-dataset
[4]https://www.nltk.org/

B. Comparison of Machine Learning Classifiers

1) SMS Classification Comparison: SMS only contains the text. We collected our SMS ham and phishing data, cleaned and pre-processed the data, implemented tokenization and feature vectorization using TFIDF, and performed model and hyperparameter tuning with GridSearch [5]. SGD classifier showed better accuracy at 98.4% performing better than SVC(98.1%).

2) Email Classification Comparison: We classified the email address, subject line, and body separately for emails.

Email Address Classification: In the test dataset, the SGD classifier performed best with an accuracy of 93.5%, and KNN followed it with 92.9%. Random Forest showed the lowest level of accuracy in predicting phishing email addresses with 84.1%. Logistic Regression and Decision Trees are closer with 85.6% and 85.4%, respectively.

Email Subject Classification: SVC was classified with a higher level of accuracy of 93.7% while SGD followed closely with an accuracy of 93.4% in the subject classification. This is followed by KNN and Gradient Boosting at 91.5% and 91.1%, respectively. Finally, random Forest exhibits the lowest accuracy level at 85.3%.

Email Body Classification: The highest level of accuracy was seen in the email body classification. SGD classifier and SVC ranked higher accuracy with 98.7% and 98.3%, respectively. This was followed by KNN and Gradient Boosting with 98.0% and 96.5%, respectively. While Random Forest (88.8%) increased accuracy compared to email addresses and email bodies, it remained the lowest.

III. PhisherCop: Overview and Performance

PhisherCop was implemented by using Python Flask Framework [6]. Users input the sender's email address, email body, and email subject or the SMS message, and the legitimacy score is calculated.

A. Web Interface

The web portal is designed with Hypertext Markup Language (HTML) [7] and Cascading Style Sheets (CSS) and styled with CSS [8]. The interface is designed with three fields and a submit button. The content field is mandatory and collects either the email body or the SMS message. However, the subject and the email address are optional, only to be inputted for email messages.

B. Input Analysis

We introduced the pickle module [9] to embed the machine learning classifiers with the highest accuracy in our web application. The module uses the dump function to embed or pickle a fitted classifier into a file or object, which can then be loaded, unpickled, or deserialized through the load function. The inputted content, email address, and subject are parsed to the unpickled classifier for the classification to determine its legitimacy.

The dataset was divided into two parts, the training dataset and the test dataset. First, we split the datasets into 70% training and 30% testing. To evaluate the performance of the classifiers, we measured the accuracy. Our experiment indicated that SGD better predicted SMS content with an accuracy score of 98.4% when fitted to the test data. This highlights that SGD performs well when used to predict the legitimacy of an SMS. Although [8] indicated that Neural Network better-predicted SMS, they only compared with SVM and Decision Tree. While SGD was more accurate than SVM and Decision Tree, we did not compare it with Neural Network.

For email classification, SGD showed higher accuracy for an email address and email body at an average accuracy of 93.5% and 98.7%, respectively. This is similar to the work of [5], where SGD accrued an accuracy of 98.1%. On the other hand, SVC gave the highest accuracy for email subject lines at an average accuracy of 93.7%.

IV. Conclusion

This paper reports on an anti-phishing detection tool that we created called PhisherCop. We implemented NLP and ML techniques to differentiate between legitimate and phishing emails and legitimate SMS and phishing SMS. We compared eight ML classifiers, and we realized the SGD classifier performed very well in detecting phishing SMS, phishing email address and phishing email bodies. Our analysis found that SVC gave higher accuracy for detecting phishing subject lines. Upon completing our comparison, we implemented PhisherCop, a web-based tool that predicts the validity of an email (subject line, email address, and body) and SMS through the Accuracy score. We leveraged the best performing classifiers we identified from our classification comparison.

V. Limitations and Future Work

We implemented NLP and ML techniques to develop an anti-phishing tool called PhisherCop. We will include BERT [3] as a baseline for our ground truth evaluation. While we focused on accuracy score which gives the ratio of correct predictions to the total number of input samples. Other evaluation metrics such as Precision, Recall, Specificity and F1 score could be explored to determine the robustness of the model. Thus, we plan to extend this work to test the users by conducting user studies with this tool as a future extension.

VI. Acknowledgement

We would like to acknowledge the Inclusive Security and Privacy-focused Innovative Research in Information Technology (InSpirit) Lab at the University of Denver. Any opinions, findings, and conclusions or recommendations expressed in this material are solely those of the authors and do and do not necessarily reflect the views of the University of Denver.

[5] https://scikit-learn.org/stable/modules/grid$_search.html$
[6] https://flask.palletsprojects.com/en/2.0.x/
[7] https://www.w3schools.com/html/
[8] https://www.w3schools.com/css/
[9] https://docs.python.org/3/library/pickle.html

REFERENCES

[1] DAS, S., ABBOTT, J., GOPAVARAM, S., BLYTHE, J., AND CAMP, L. J. User-centered risk communication for safer browsing. In *International Conference on Financial Cryptography and Data Security* (2020), Springer, pp. 18–35.

[2] DAS, S., NIPPERT-ENG, C., AND CAMP, L. J. Evaluating user susceptibility to phishing attacks. *Information & Computer Security* (2022).

[3] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] DHAMIJA, R., TYGAR, J. D., AND HEARST, M. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (2006), pp. 581–590.

[5] ELSHOUSH, H. T., AND DINAR, E. A. Using adaboost and stochastic gradient descent (sgd) algorithms with r and orange software for filtering e-mail spam. In *2019 11th Computer Science and Electronic Engineering (CEEC)* (2019), IEEE, pp. 41–46.

[6] MISHRA, S., AND SONI, D. Sms phishing and mitigation approaches. In *2019 Twelfth International Conference on Contemporary Computing (IC3)* (2019), IEEE, pp. 1–5.

[7] MISHRA, S., AND SONI, D. Dsmishsms-a system to detect smishing sms. *Neural Computing and Applications* (2021), 1–18.

[8] SHAHI, T. B., AND SHAKYA, S. Nepali sms filtering using decision trees, neural network and support vector machine. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)* (2018), IEEE, pp. 1038–1042.

[9] UNCHIT, P., DAS, S., KIM, A., AND CAMP, L. J. Quantifying susceptibility to spear phishing in a high school environment using signal detection theory. In *International Symposium on Human Aspects of Information Security and Assurance* (2020), Springer, pp. 109–120.