e-ISSN:2582-7219

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

Volume 7, Issue 10, October 2024

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.521

© 2024 IJMRSET | Volume 7, Issue 10, October 2024| DOI: 10.15680/IJMRSET.2024.0710063

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Machine Learning-Powered Chrome Extension for Advanced Email Phishing Detection

D.Sterlin Rani, Mettu Yamini, Mothukuri Chandanasri, Naga Krishna Priya

Faculty, Department of Computer Science and Engineering, R.M.D. Engineering College, Chennai, India

U.G. Student, Department of Computer Science and Engineering, R.M.D. Engineering College, Chennai, India

U.G. Student, Department of Computer Science and Engineering, R.M.D. Engineering College, Chennai, India

U.G. Student, Department of Computer Science and Engineering, R.M.D. Engineering College, Chennai, India

ABSTRACT: The use of cybersecurity is all about protecting our digital world. It also involves servers, phones, computer, network safe from attacks by people who want to steal the information, cause damage, or disrupt operations. It's look like having a strong lock on your front door to keep burglars out, but for your data and digital devices. This study examines how effective persuasion cues, specifically gain and loss, are in detecting phishing emails. Phishing is a malicious attempt to obtain sensitive information from unsuspecting individuals. Our project develops a phishing detection system using machine learning, seamlessly integrated as a Chrome extension. With feature selection, algorithm evaluation, and model optimization, the system ensures accurate detection of phishing emails. Real-time protection is provided within user's email experience, validated through rigorous testing and user feedback, enhancing overall email security. The primary expected result is a significant reduction in false negatives. Traditional security filters may overlook subtle linguistic cues that characterize phishing emails, leading to undetected threats. With NLP, the system becomes adept at recognizing anomalies in language usage, enhancing its capacity to flag suspicious emails accurately. This reduction in false negatives is pivotal in preventing successful phishing attacks and safeguarding sensitive information.

KEYWORDS: Phishing, Malicious, Natural Language Processing, Cyber Security

I. INTRODUCTION

Social engineering, a cunning cyber-attack tactic, exploits human psychology and unsuspecting behaviors to dupe victims into divulging sensitive information or performing fraudulent actions. One prevalent form of this deceit is phishing, where in perpetrators craft counterfeit web pages resembling legitimate ones to trick users. Phishing stands as a sinister blend of social engineering and technical deceit, aiming to pilfer personal data and financial credentials. However, traditional methods like blacklisting URLs fall short in detecting novel phishing pages. To counter this, Machine Learning (ML), particularly Deep Learning (DL), emerges as a potent solution. DL models outshine traditional ML in accuracy, albeit at the cost of increased computation time. In phishing page analysis, Natural Language Processing (NLP) plays a pivotal role, decoding human language nuances. DL approaches often confront the challenge of non-sequential text input, undermining semantic coherence. Yet, employing sequential methods like Keras Embedding with GloVe preserves textual context, fostering semantic and syntactic relationships between words. (18). The Anti-Phishing Working Group (APWG) is an organization dedicated to combating online phishing scams by collecting, analyzing, and sharing information about fraudulent websites that attempt to steal sensitive information from users. They release quarterly reports detailing the prevalence (7).

Traditional ML algorithms often provide more interpretable results compared to deep learning models. While interpretability is not directly related to computation time, the additional complexity of deep learning models might make interpretation more challenging, requiring additional computational resources for post-processing or analysis (18).

II. RELATED WORK

This survey examines the use of Natural Language Processing (NLP) and Machine Learning (ML) techniques to detect phishing emails, highlighting state-of-the-art strategies and future research directions[1]. This paper proposes an NLP and ML-based approach for detecting phishing emails, comparing six ML algorithms and demonstrating high accuracy, precision, and recall on two public datasets, offering insights for improved phishing attack prevention. Prepare Your Paper Before Styling[2]. This research introduces a novel deep learning approach, combining ResNeXt and GRU models with SMOTE for data imbalance, achieving superior phishing attack detection accuracy of 98% and outperforming state-of-the-art algorithms by 11% to 19%, enhancing digital forensics and cybersecurity [3]. This study proposes a novel approach using Natural Language Processing and Deep Learning algorithms, achieving high accuracy in detecting phishing attacks based on the text of suspicious web pages, with the best performer, Bidirectional GRU, achieving 97.39% accuracy [4].

This paper reviews and analyses e-mail classification articles from 2006-2016, exploring application areas, datasets, features, classification techniques, and performance measures, and identifies research directions and challenges for future work [5].This study conducted a large-scale anti-phishing training, analysing user click behaviour and developing a novel machine learning model to predict phishing susceptibility, highlighting the importance of individualized training over group training [6]. This paper provides a comprehensive survey of recent advancements in machine learning-based encrypted traffic analysis, discussing goals, methodologies, and challenges, with a focus on network asset identification, network characterization, privacy leakage detection, and anomaly detection [7].

This study proposes a phishing email classifier model using deep learning and graph convolutional networks (GCN) combined with natural language processing, achieving a high accuracy rate of 98.2% and a low false-positive rate of 0.015 in detecting phishing emails based on body text [8]. This paper introduces a natural language-based scheme, PhishNet- NLP, to detect phishing emails by utilizing features that characterize phishing, focusing on distinguishing between "actionable" and "informational" emails, and leveraging natural language techniques and contextual information to outperform existing detection schemes [9]. This paper proposes a new framework and a comprehensive model for detecting and defending against phishing attacks, utilizing a hybrid multi-layer model with Natural Language Processing (NLP) techniques to characterize the behaviour of the attacks [10].

This paper proposes a multi-stage approach using natural language processing and machine learning to detect phishing email attacks, achieving a 100% success rate in F1-measure with reduced feature sets and lower computational cost compared to state-of-the-art schemes [11].This study proposes a hybrid machine learning classifier using TF-IDF and a feature extraction technique to detect phishing emails, achieving an accuracy of 87.5% on a real-world dataset and highlighting the effectiveness of combining different models for improved performance [12].

This research paper focuses on using natural language processing (NLP) concepts for classifying phishing emails, discussing various technologies of phishing, evaluation metrics, literature review, and the solution approach with detailed descriptions of NLP concepts and working procedures [13]. This paper introduces the DARTH framework, which utilizes machine learning and NLP techniques to accurately identify phishing emails by analysing multiple composite features, achieving high precision and accuracy rates [14]. This paper presents a real-time anti-phishing system using seven classification algorithms and NLP-based features, achieving a 97.98% accuracy rate in detecting phishing URLs, with distinguishing features including language independence, use of a large dataset, and real-time execution [15].

This paper introduces a novel approach to phishing detection by analysing hyperlinks in HTML source code, achieving over 98.4% accuracy using logistic regression, with features categorized into 12 types and offering language independence and client-side implementation [16]. These abstract highlights the role of natural language processing (NLP) in detecting complex phishing emails, utilizing semantic analysis, sentiment analysis, name recognition, and natural language generation to provide robust protection against digital deception [17].

This paper presents a comprehensive survey of phishing detection techniques, focusing on Machine Learning (ML) and Nature-Inspired (NI) approaches, highlighting the need for more efficient and reliable solutions to tackle evolving

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

phishing schemes [18]. This chapter examines the impact of feature selection methods on efficiently and effectively detecting phishing webpages, demonstrating that using these methods with machine learning algorithms can significantly improve model building time without compromising accuracy [19]. This study investigates the effectiveness of federated learning (FL) in phishing email detection, showing comparable performance to centralized learning for balanced datasets and low organizational counts, but observing variation in performance with increased organizational counts and dataset asymmetry [20].

III. DATASET DESCRIPTION

1. **Attachments:** Indicates whether an email contains attachments.
2. **Malicious URL**: Identifies if the email includes a URL that is potentially malicious.
3. **Text link disparity**: Measures the discrepancy between the text displayed for a link and its actual URL.
4. **Number of dash:** Counts the number of dash characters in the email.
5. **Mailto:** Presence of "mailto:" links in the email.
6. **HTML:** Indicates whether the email contains HTML formatting.
7. **IP URLs**: Presence of URLs with IP addresses.
8. **Contains account:** Identifies if the email contains references to user accounts.
9. **Re: mail:** Presence of email replies indicated by "Re:" in the subject line.
10. **Maximum Domains Counts:** Counts the maximum number of unique domains in the email.
11. **Number of URLs:** Total count of URLs in the email.
12. **Hexadecimal URL:** Presence of URLs using hexadecimal encoding.
13. **General Salutation:** Presence of general salutations (e.g., "Dear Sir/Madam").
14. **Body richness:** Measures the richness of the email body content.
15. **Number of dots:** Counts the number of dot characters in the email.
16. **Contains prime targets:** Identifies if the email contains references to high-value targets (e.g., executives or sensitive information).

	Model	Accuracy	Balanced Accuracy	Log Loss	F1 Score	MCC
0	Logistic Repression (before turning)	95.100	95.100	0.205	95.122	0.902
1	Logistic Repression (after tuning)	97.090	97.087	0.162	97.027	0.943
2	SVC (before tuning)	94.793	94.794	0.139	94.801	0.896
3	SVC (after tuning)	97.090	97.091	0.11	97.099	0.942
4	Gradient Boosting Clssifier (before tuning)	97.550	97.549	0.093	97.538	0.951
5	Gradient Boosting (after tuning)	98.009	98.008	0.072	97.991	0.960
6	ExtraTrees Classifier (before tuning)	97.550	97.550	0.092	97.546	0.951
7	ExtraTrees Classifier (after tuning)	98.469	98.468	0.071	96.462	0.969
8	Random Forest Classifier (before tuning)	98.315	98.315	0.143	98.310	0.966
9	Random Forest Classifier (before tuning)	98.469	98.468	0.066	98.457	0.969
10	Voting Classifier (hard voting)	97.550	97.550	N/A	97.546	0.951
11	Voting Classifier (soft voting)	97.243	97.244	0.094	97.248	0.945

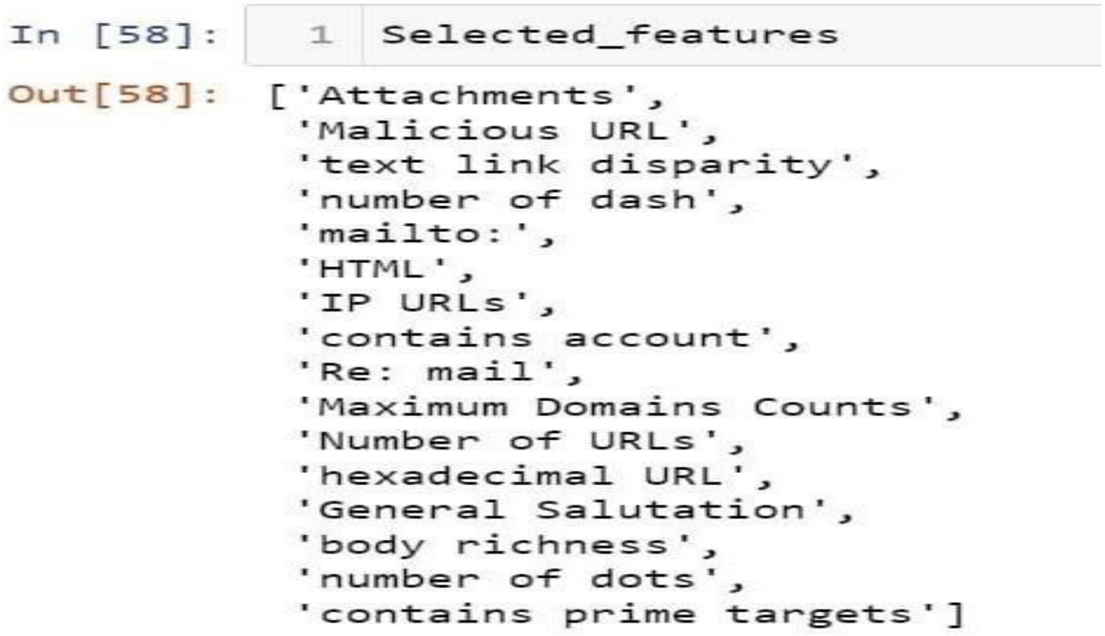
Table 1: Dataset

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. FEATURES



```
In [58]:  Selected_features

Out[58]: ['Attachments',
           'Malicious URL',
           'text link disparity',
           'number of dash',
           'mailto:',
           'HTML',
           'IP URLs',
           'contains account',
           'Re: mail',
           'Maximum Domains Counts',
           'Number of URLs',
           'hexadecimal URL',
           'General Salutation',
           'body richness',
           'number of dots',
           'contains prime targets']
```

Fig. 1. Features

- Attachments
- Malicious URL
- Text link disparity
- Number of dash
- Mailto
- HTML
- IP urls
- Contains account
- Re: mail
- Maximum Domains Counts
- Number of urls
- Hexadecimal URL
- Body richness
- Number of dots
- Contains prime targets

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. SYSTEM DESIGN

In implementing a phishing email detection system, the process typically involves collecting a diverse dataset of phishing and non-phishing emails. After visualizing the data, relevant features like URL and mail characteristics are extracted.

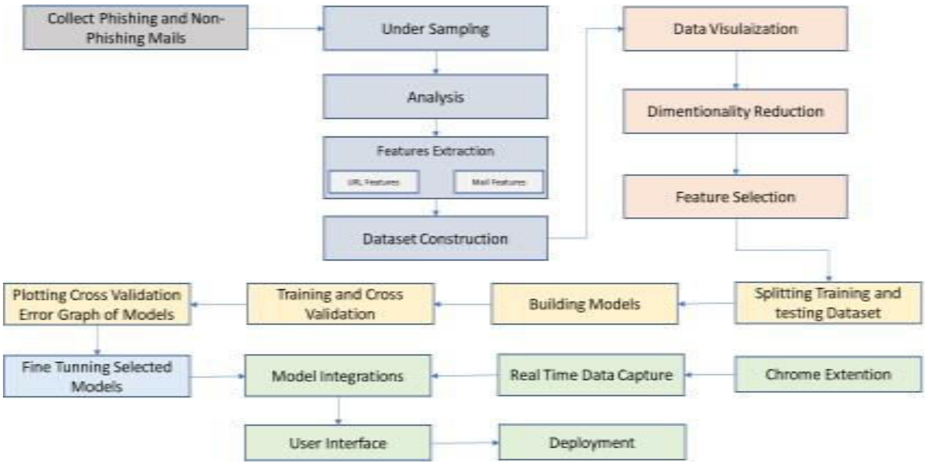


Fig : 2 Block Diagram

In implementing a phishing email detection system, the process typically involves collecting a diverse dataset of phishing and non-phishing emails. After visualizing the data, relevant features like URL and mail characteristics are extracted.

Dimensionality reduction and feature selection techniques are applied to streamline the dataset. Following the construction of a structured CSV file, the dataset is split into training and test sets. Model selection and fine-tuning, informed by cross- validation, contribute to building effective classification models. The cross-validation error graph aids in evaluating and selecting the best model. Integration of models under a voting classifier enhances overall performance, culminating in a comprehensive performance comparison on the test dataset. This iterative approach ensures a robust and accurate phishing email detection system.

VI. IMPLEMENTATION

Mail Reception Module-The module is initialized with the paths to the client secret and token files. These files are essential for authenticating the application with the Gmail API. The authenticate method is responsible for loading existing credentials from the token file or creating new ones if needed. It checks if the credentials are valid, refreshing them if expired. If the credentials don't exist, it initiates the OAuth2 authentication flow and saves the new credentials to the token file. The connect_to_gmail_api method builds a connection to the Gmail API using the authenticated credentials. It returns the Gmail API service object, which will be used to interact with Gmail. The monitor inbox method calls connect_to_gmail_api to get the Gmail API service. It then uses the Gmail API to monitor the user's inbox for new emails. This involves using the users(), messages(), list() method to retrieve a list of messages. The code snippet for handling new emails is marked with a comment. This part should be implemented based on your specific requirements, such as fetching email details or triggering actions based on new emails. It catches and prints any errors that might occur during the monitoring process. In the main block, an instance of the Mail Reception Module is created. The monitor inbox method is called to initiate the process of monitoring the inbox for new email. Throughout the code, there are error-handling mechanisms using try-except blocks. These ensure that if there's an issue with authentication, connecting to the Gmail API, or monitoring the inbox, the application can gracefully handle and report the error. Data Extraction Module-The raw email message is typically in MIME (Multipurpose Internet Mail Extensions) format. Parsing involves breaking down the email into its different components, such as headers, body, and attachments.

Python libraries like email can be used to parse raw email messages. Metadata in email includes information about the email itself, like the sender, recipients, subject, date, and any other relevant details. The email library can help extract metadata from email headers and other sections. Email headers contain important information about the email's origin, routing, and authentication status. Headers like From, To, Subject, Received, and Authentication-Results can be parsed to gather relevant details. SPF (Sender Policy Framework) and DKIM (DomainKeys Identified Mail) headers can be checked for authentication status. The email body contains the actual content of the message, which can be in plain text, HTML, or both. The email library can be used to extract the text and HTML content of the email body. Embedded links in the email body are typically found in HTML content. Parsing the HTML content and using libraries like Beautiful Soup can help extract and retrieve links. It's essential to validate and check the legitimacy of these links, as they could be potentially harmful. Attachments can include files or embedded images. The email library can assist in extracting and saving attachments. Special attention should be given to security measures to prevent executing malicious code from attachments.

Feature Extraction Module-NLTK is a important Python library for working with mortal language data. It provides tools for tasks similar as tokenization, stemming, part- of- speech trailing, and more. Tokenization involves breaking down textbook into individual words or rulings, making it easier to dissect. Dispatch Content Analysis for verbal Patterns The dispatch content is reused using NLTK to identify verbal patterns. This could include rooting important words, expressions, or realities. NLTK's capabilities can be employed to dissect the syntactic and semantic structure of the textbook. Keyword birth Keywords are specific words or expressions that capture the substance of the textbook. NLTK can be used for keyword birth by considering the frequence and significance of words within the dispatch content. ways like TF- IDF (Term frequencies- Inverse Document frequency) can be employed to assign weights to words grounded on their significance in the environment of the entire dispatch corpus. Sentiment Analysis NLTK provides tools for sentiment analysis, allowing you to determine the sentiment (positive, negative, or neutral) expressed in the dispatch. Sentiment analysis can be useful in understanding the emotional tone of the communication. Identification of Structural and Content- Grounded Features Structural features may include the length of the dispatch, the presence of specific rudiments like felicitations or autographs, and the association of paragraphs. Content- grounded features may involve the identification of specific motifs, the operation of certain language constructs, or the presence of attachments. Generation of point Vector for Machine Learning The uprooted features (keywords, sentiment, structural and happy- grounded features) are used to produce a point vector for each dispatch. This point vector serves as input to a machine literacy model, allowing the model to learn patterns and make prognostications grounded on the linked features. Legitimacy Validation Module-Begin by collecting a dataset of emails, labeled with their corresponding orders (spam or not spam). Excerpt applicable features from the emails, similar as word frequentness, presence of specific keywords, sender information, and other applicable metadata. transfigure these features into a numerical format, creating a point vector for each dispatch. Classifier perpetration Choose a machine learning algorithm for bracket. Common choices for dispatch analysis include decision trees, arbitrary timbers, support vector machines, or grade boosting classifiers. Split your dataset into training and testing sets to estimate the performance of your model. Ensemble Learning apply ensemble literacy ways to combine the prognostications of multiple base classifiers. This can ameliorate the overall delicacy and robustness of the model. exemplifications of ensemble styles include Random timbers, AdaBoost, and Gradient Boosting Machines. point significance Analysis After training your classifier, dissect the significance of each point. Some models, like Random timbers, give point significance scores. Identify the most influential features, as this information can be precious for understanding the decision- making process of your model.

Optimization ways Fine- tune hyperparameters of your chosen classifier to enhance its performance. This can be done through ways like grid hunt or arbitrary hunt. Use cross-validation to insure the conception of your model and avoid overfitting. Hyperparameter Tuning with tools like HyperRFC- Tuner HyperRFC- Tuner, as an illustration, could be a tool that automates the process of hyperparameter tuning specifically for Random Forest classifiers. This tool might perform a total or randomized hunt over a predefined hyperparameter space to find the optimal configuration for your Random Forest model. By using similar tools, you can save time and coffers in chancing the stylish hyperparameters for your specific problem. Evaluation and Deployment estimate the performance of your model on a separate test set. Common criteria for bracket tasks include delicacy, perfection, recall, and F1 score. Once satisfied with the performance, emplace the model to dissect incoming emails and classify them as spam or not spam.

Result Display Module-The system evaluates incoming emails and classifies them into categories, typically distinguishing between "spam" and "non-spam" (ham) emails. The result of this classification is then presented to the user, often with visual indicators such as labels, icons, or color codes. For instance, spam emails might be marked with a red icon, while non-spam emails could have a green icon. After presenting the classification result, the system provides the user with actionable options. Common actions include. Users can manually label an email as spam, helping improve the system's future classifications. Users may choose to delete unwanted emails directly. In the case of false positives (non-spam emails classified as spam), users can move the email to the inbox to correct the classification. Report as Phishing: If the system includes phishing detection, users may be able to report suspicious emails. The system ensures a seamless integration with the email client's interface, meaning that these classification results and user actions are presented in a way that aligns with the email client's design and functionality. This could involve using standard UI elements, such as buttons, checkboxes, and context menus, to maintain consistency with the overall email client experience. The system may offer customization options to cater to individual user preferences. This could include users might be able to choose how the classification results are displayed, such as the size and color of icons or labels. Users may customize default actions, like whether marked spam emails are automatically deleted or moved to a designated folder. The system might incorporate a feedback loop where user actions (such as marking an email as spam) contribute to the improvement of the machine learning model. This helps refine the system's accuracy over time.

VII. RESULT AND DISCUSSION

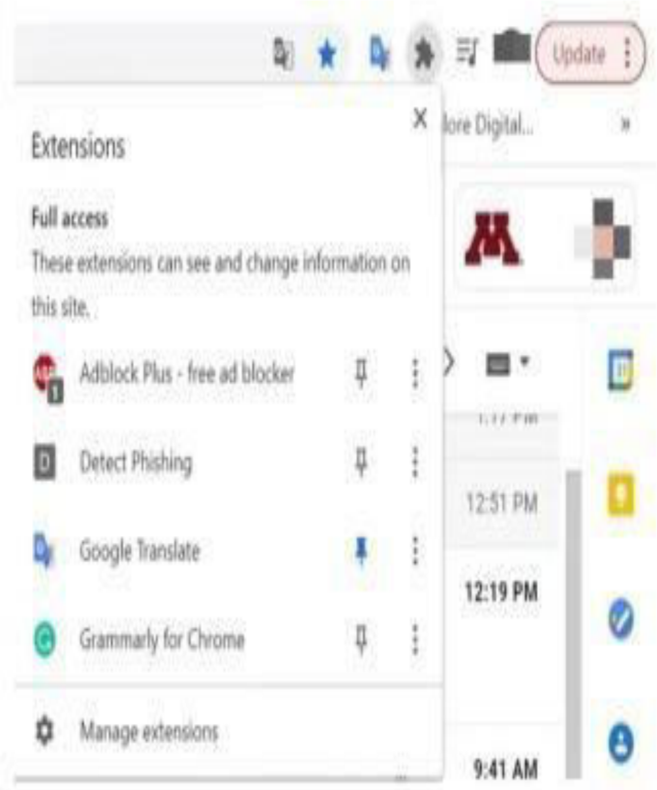
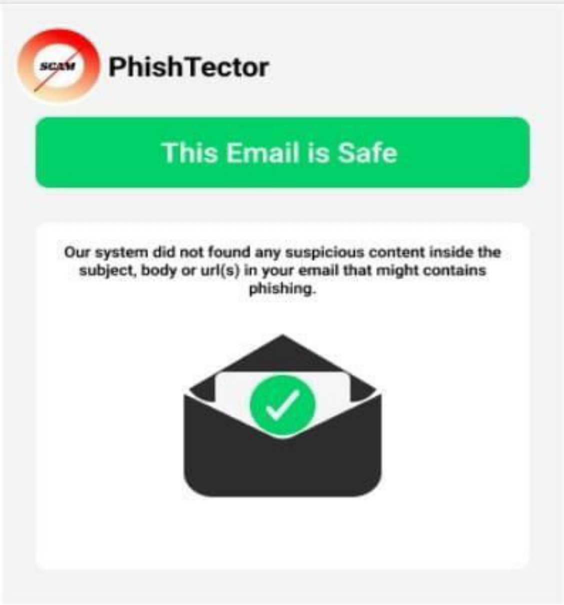


Fig. 3. Chrome Extension

The primary anticipated result is a significant reduction in false negatives. Traditional security pollutants may overlook subtle verbal cues that characterize phishing emails, leading to undetected pitfalls. With NLP, the system becomes complete at feting anomalies in language operation, enhancing its capacity to flag suspicious emails directly. This reduction in false negatives is vital in precluding successful phishing attacks and securing sensitive information.

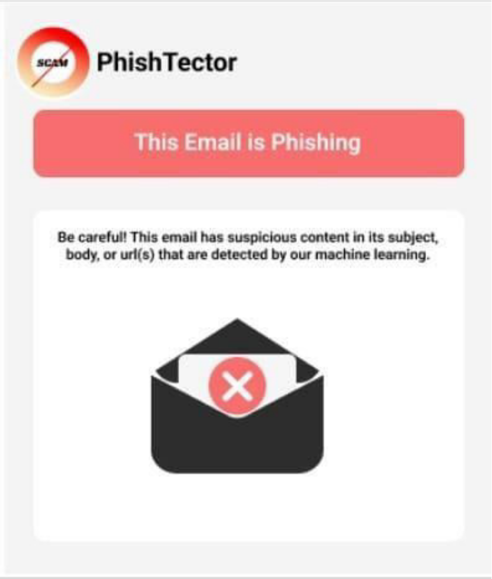


PhishTector

This Email is Safe

Our system did not found any suspicious content inside the subject, body or url(s) in your email that might contains phishing.

Fig. 4. Showing that mail is safe



PhishTector

This Email is Phishing

Be careful! This email has suspicious content in its subject, body, or url(s) that are detected by our machine learning.

Fig. 5. Showing that the mail is not safe.

VIII. CONCLUSION

In conclusion, the integration of Natural Language Processing into email security systems represents a significant stride towards mitigating the growing threat of phishing attacks. By harnessing the power of linguistic analysis and contextual understanding, NLP not only enhances the accuracy of detection but also enables organizations to adapt to the evolving tactics of cyber adversaries. As the digital landscape continues to evolve, leveraging advanced technologies such as NLP is essential for safeguarding the integrity and confidentiality of email communications.

REFERENCES

[1] Thapa, C., Tang, J.W., Abuadbba, A., Gao, Y., Camtepe, S., Nepal, S., Almashor, M. and Zheng, Y., 2023. Evaluation of federated learning in phishing email detection. *Sensors*, *23*(9), p.4346. *(references)*

[2] Gandotra, E. and Gupta, D., 2021. An efficient approach for phishing detection using machine learning. *Multimedia Security: Algorithm Development, Analysis and Applications*, pp.239-253.

[3] Akinyelu, A.A., 2019. Machine learning and nature inspired based phishing detection: a literature survey. *International Journal on Artificial Intelligence Tools*, *28*(05), p.1930002.

[4] REDDY, K.T., NLP in Cybersecurity: Analyzing Phishing Emails for Enhanced Protection.

[5] Jain, A.K. and Gupta, B.B., 2019. A machine learning based approach for phishing detection using hyperlinks information. *Journal of Ambient Intelligence and Humanized Computing*, *10*, pp.2015-2028.

[6] Sahingoz, O.K., Buber, E., Demir, O. and Diri, B., 2019. Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, pp.345-357.

[7] Mittal, A., Engels, D.D., Kommanapalli, H., Sivaraman, R. and Chowdhury, T., 2022. Phishing Detection Using Natural Language Processing and Machine Learning. SMU Data Science Review, 6(2), p.14.

[8] Verma, P., Goyal, A. and Gigras, Y., 2020. Email phishing: Text classification using natural language processing. *Computer Science and Information Technologies*, *1*(1), pp.1-12.

[9] Palanichamy, N. and Shri Murti, Y., 2023. Improving phishing email detection using the hybrid machine learning approach. *Journal of Telecommunications and the Digital Economy*, *11*(3), pp.120-142.

[10] Gualberto, E.S., De Sousa, R.T., Vieira, T.P.D.B., Da Costa, J.P.C.L. and Duque, C.G., 2020. The answer is in the text: Multi-stage methods for phishing detection based on feature engineering. *IEEE Access*, *8*, pp.223529-223547.

[11] Thakur, K., Shan, J. and Pathan, A.S.K., 2018. Innovations of phishing defense: The mechanism, measurement and defense strategies. *International Journal of Communication Networks and Information Security*, *10*(1), pp.19-27.

[12] Verma, R., Shashidhar, N. and Hossain, N., 2012. Detecting phishing emails the natural language way. In Computer Security–ESORICS 2012: 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10-12, 2012. Proceedings 17 (pp. 824-841). Springer Berlin Heidelberg.

[13] Alhogail, A. and Alsabih, A., 2021. Applying machine learning and natural language processing to detect phishing email. *Computers & Security*, *110*, p.102414.

[14] Gambín, Á.F., Yazidi, A., Vasilakos, A., Haugerud, H. and Djenouri, Y., 2024. Deepfakes: current and future trends. *Artificial Intelligence Review*, *57*(3), p.64.

[15] Shen, M., Ye, K., Liu, X., Zhu, L., Kang, J., Yu, S., Li, Q. and Xu, K.,
2022. Machine learning-powered encrypted network traffic analysis: a comprehensive survey. *IEEE Communications Surveys & Tutorials*.

[16] Sutter, T., Bozkir, A.S., Gehring, B. and Berlich, P., 2022. Avoiding the hook: influential factors of phishing awareness training on click- rates and a data-driven approach to predict email difficulty perception. *IEEE Access*, *10*, pp.100540-100565.

[17] Gupta, S. and Kumaraguru, P., 2014, September. Emerging phishing trends and effectiveness of the anti-phishing landing page. In *2014 APWG Symposium on Electronic Crime Research (eCrime)* (pp. 36- 47). IEEE.

[18] Benavides-Astudillo, E., Fuertes, W., Sanchez-Gordon, S., Nuñez- Agurto, D. and Rodríguez-Galán, G., 2023. A Phishing-Attack- Detection Model Using Natural Language Processing and Deep Learning. *Applied Sciences*, *13*(9), p.5275.

[19] Alsubaei, F.S., Almazroi, A.A. and Ayub, N., 2024. Enhancing phishing detection: A novel hybrid deep learning framework for cybercrime forensics. *IEEE Access*.

[20] Alsubaei, F.S., Almazroi, A.A. and Ayub, N., 2024. Enhancing phishing detection: A novel hybrid deep learning framework for cybercrime forensics. IEEE Access.

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH
IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com