

Phishing Attack Detection using NLP and Machine Learning

Naresh Kamble¹, Dr. Nilamadhab Mishra² and Sonali Rokade³

¹⁻³School of Computer Science and Engineering, VIT Bhopal University Kothrikalan, Sehore, Madhya Pradesh India
Email: naresh.kamble2019@vitbhopal.ac.in, nilamadhab.mishra@vitbhopal.ac.in, rokade.shashikant2019@vitbhopal.ac.in

Abstract— It's observed that in recent years, cyber-attacks are becoming a regular threat. In conventional phishing attacks, targeted users are redirected to visit a fake or counterfeit website, for taking out personal information and credentials like banking information, credentials, and even UPI wallets. Usually, phishing attacks happens using email, text messages, and voice calls by forwarding fake website URL. This attack is not only referring to users who have very little information about this fraud but also those who have good and sound knowledge of this type of attack. Detection of a Social engineering attack such as phishing is a challenging task due to its nature of affecting. In this research, the System uses the approach of detecting this sophisticated attack using ML, NLP, WDM, and MCAM which is the best suitable approach to identify the phishing attack. Machine Learning is the most advanced way of detecting targeted phishing attacks from URL, Email, or SMS. a Detection is done with NLP and Machine learning with attack vectors as a dataset with help of simulators performing data visualization work.

Keywords--Natural Language Processing (NLP), Machine learning (ML), CNN, Phishing attack, WDM, MCAM

I. INTRODUCTION

Advancement in the sector of communication and networks, the Internet is the most dominating area for exchanging information between communication nodes and networks. Human life is made easy by digitalization and the use of digital media for internet banking, E-mail, communication, phone calls, and video transmission made it useful in the personal as well as in the professional life of the individual. Nowadays everyone is connected to the internet or network by some means. Use of these technologies no doubt created ease in working but involving too much information exposed the risk of Cyber Attacks and Credential Hijacking. Critical Vulnerabilities may give remote access to the computer or computer network to Cyber Attackers and then these attackers can do anything they want using these systems.

The effect of this type of security vulnerability increases the threat of financial loss over banking channels. Inexperienced users may fall into these traps and lose their hard-earned wealth or reputation. Cyber-attacks cause a huge loss of private data and even the wealth of professionals, companies, and individuals daily. The data which is lost in these attacks are in fact of billions of dollars annually. Phishing is a major source of these financial attacks. Cyber attackers create a counterfeit website having the same design and look as the actual one and spread it using the most preferred and famous communication channels such as Instant messengers, SMS, and Email. These especially started with fake emails,

When users connect to a site that appears to be genuine, they enter the chosen information without the knowledge

that the site is fake. The attacker usually forwards the user to a fake website. A similar instance of the most recent situation is shown in Figure. 1. Shows the website from the link sent in the e-mail which appears as a masked link. When the redirected page is open, we can see the spoofed website which is the exact copy of the famous e-commerce website. All these efforts are made by the attacker to make sure the user will feel that it is the actual website and forced to enter login information or banking information. Similar links are sent out on SMS stating the company is providing a 90% discount on all products listed and today is the last day. Once a user clicks on the link in the SMS, they likely get this type of fake or carefully created webpages. Even though the different website address is visible, an uninformed user will not notice that it is a trap.

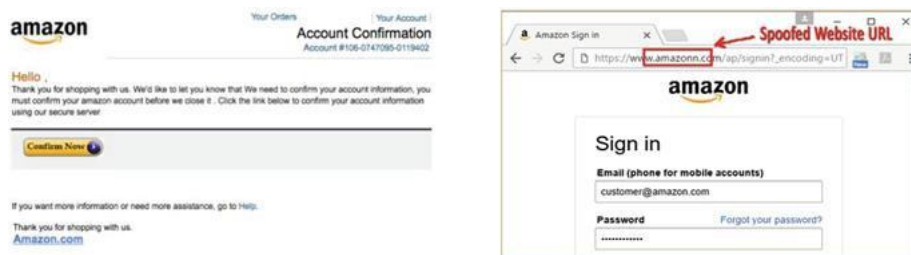


Figure 1. Email received with Phishing Page

A Lot of studies in detection of the Phishing Attacks have been conducted. Natural Language Processing, CNN, Machine Learning, and Deep Learning are the main focus. Recognition of phishing from the website address can be an immaterial task for many specialists since the attacker can trick cyber-literate clients with various professional and sophisticated web methods. The easiest is to use of clone website with similar code or an exact copy of the original website. These attacks are carried out with all precautions and expert modes and therefore, the detection of such attacks with the application is important and a multi-approach is required since a single approach is not adequate for a proper detection mechanism. For this purpose, hybrid detection methods are preferred. The website name, Domain-Names, keywords, and FAQs are used to detect the attack using Natural Language Processing (NLP). We have figured out the exact characteristics for the detection of such attacks and evaluated the effect using only the Concurrent Neural Network. The features it stored on its own are not enough to decide whether a data set is fraudulent but it will provide the information which is useful in taking the right predictive decision.

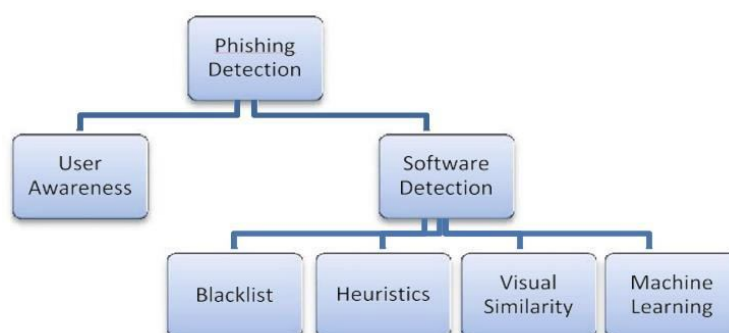


Figure 2. Structure of proposed phishing detection approaches

II. URL AND ATTACK TECHNIQUES

As a result, the attackers are using some different methods to ensure that phishing pages will not be detected and are continuously making cover pages. They make sure to use safety mechanisms or users to test their attack launch pads and then the attack is released. We use the parts in the URL (Uniform Resource Locator) structure shown in Figure 3 as a reference to the parts and the key finding for our dataset and feeding key set to NLP Model. Mainly, the attacker uses similar domain names with different domains to trick users to fall into their trap. For example, <https://www.example.com> is an actual domain name and <https://www.example.co> is the phishing page domain name. If we carefully see the domain used, we can make out the difference in the site address. This part of the URL can only be set once.

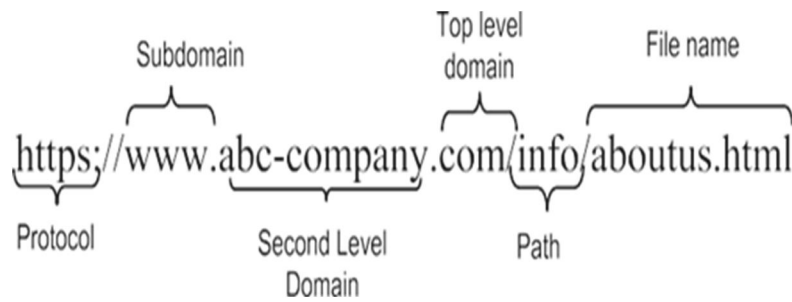


Figure. 3. Listing the components of the URL

This domain name is the main part of the detection. If we detect it, we can track the phishing. So, Information Security professionals are making a lot of effort to identify these types of domains. When a domain name is captured, it is relatively easy to block access to that counterfeit domain name and URL.

III. PREPROCESSING AND DATA SETS OF THE PARAMETERS

To create an effective phishing detection engine using Machine Learning and Natural Language Processing, we required two sets of the URL or key finding keywords, one which is a legitimate URL and another that would be fake or have a similar look to the original URL. It is important to bring out the notable features. The dangerous or targeted URLs may contain known domain names as the actual website. For the model to be able to detect a usage brand, domain names, and keywords are needed and we have created these lists using different sources.

For detecting the URL-based phishing attempts, we used different techniques to acquire URL data and took an impart from Phish Tank, which is an open repository website for possible phishing URLs and other key findings parameters which we will incorporate in our functioning system. Since the fake URLs are not high-ranking in any search engines, we used Weka as a feature extraction tool. [8,34]

The parts of a URL are separated using unique symbols. E.g, the domain name, and the domain are distinguished using a dot. Likewise, the domain and the sub-domain are also separated by a dot. Separation of the file formatting or navigation is shown using “/”.

Each URL will have its separation mark either a dot or /.

For example, “-” in the domain “example-website.com”, makes the separation. The separation can be done using “=,? &” characters. Before we started the URL pre-processing, each word was added to the list of words that is going to be analyzed. The data preprocessing flow diagram is given in the figure.

The main objectives of data pre-processing:

- To detect words similar to keywords preset with the domain name in a given URL
- To identify keywords in the URL or similarities
- To identify words with matched or random characters

In the system where RWDM has extracted the brand name and keywords the remaining words which will be analyzed from the Random Word Detection Module.

The RWDM verifies whether the words are composed of some random characters or not. Then detected words are stored in a data set word file which will be used for further analysis. When the system approached to select keywords, which are highly used in phishing it was resolute that the measurement of the words and phrases including the contiguous writing.

Therefore, the heuristic-based threshold is strongminded to detect the closest words. The words with more than 7 letters in length have been checked in Word Decomposer Module (WDM) to separate.

A. The Word Decomposer Module

The Word Decomposer Module also called WDM is responsible for the processing of words as different objects if there is more than one word written in a given character string. In certain conditions, if no adjacent words are found for the analyzed word, then the main word is returned as a parameter. Refer to the flow diagram which shows the inner working of the Word Decomposer Module.

In the dictionary of the terms, the string which is in progress is checked for possibilities of keywords that can be flagged as malicious. Later on, the term found in the dictionary is added to the word dataset. The system can distinguish contiguously written English words. Enchant packages are in use which are developed in Python to check if a word is found in the English dictionary. Then the main string is divided into characters. For example,

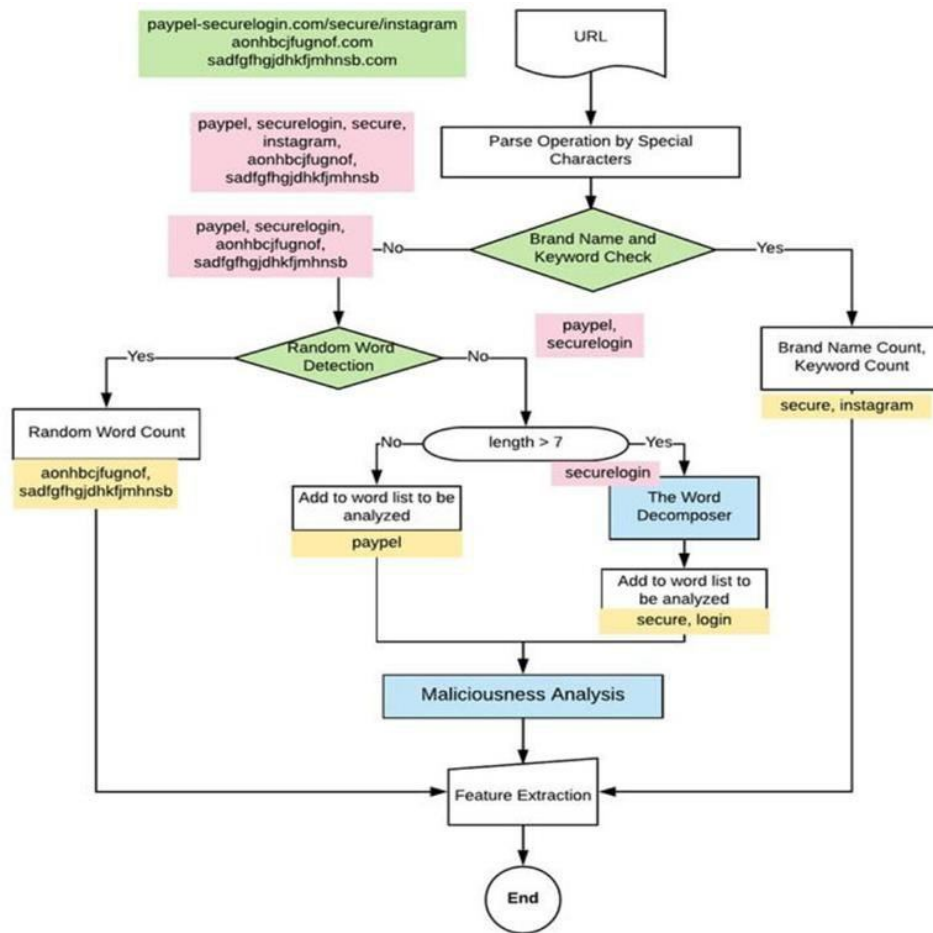


Figure. 4. The flow diagram

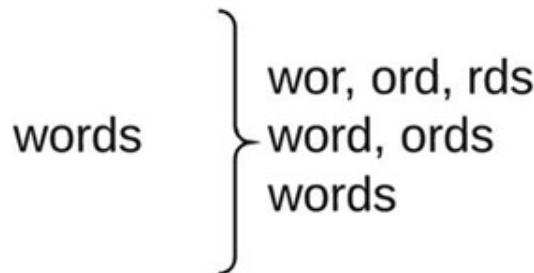


Figure. 5. Word recognition example

B. The Randomized Word And Pattern Detection Module (Rwpdm)

RWPD is the source library that helps to create random words i.e., noise in text data. The same is useful in the generation of random authorization, token generation of constant or variable length, text data augmentation, etc. To identify a random word, a consecutive letter in the word is erased during the test process. [43,56]. To decide whether the word is random, a threshold value is determined. If the multiplication of the likelihoods of the letter pairs is below this threshold value, the word is classified as random

C. Malicious Content Analysis Module (MCAM)

Developed MCAM which is a functional system inside our proposed system architecture to detect the selected word used in any phishing attacks or fraudulent purposes. This module has been used to analyze the words that need to be analyzed.

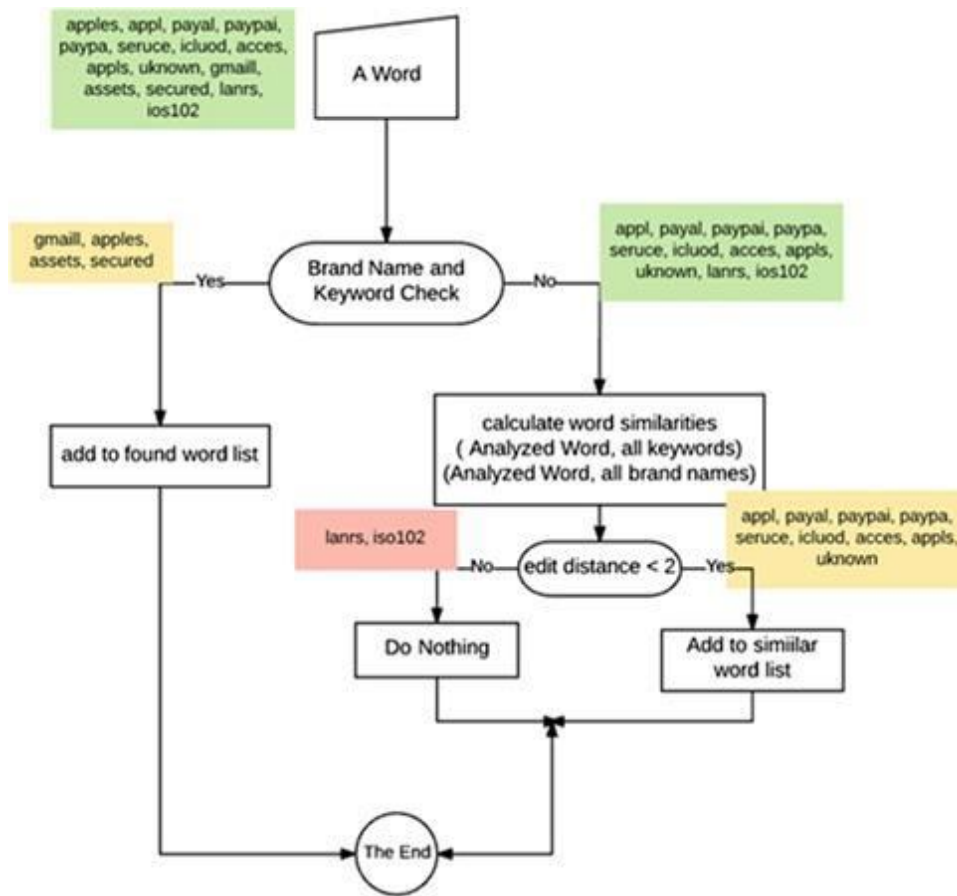


Figure. 6. Module Flow

WDM-generated words, which include the brand names or keywords, are checked by the name of the domain, the name of the brand, and the lists of the keywords in this module again.

Then, the analyzed word is compared with all the words in the name of the brand and the lists of keywords to calculate the similarity. Transactions used for conversion are the addition of, deletion of, and changing of characters.

IV. PSEUDOCODE OF THE ALGORITHM AND FRAMEWORK

TABLE I : NOTATIONS USED IN THE ALGORITHMS

Notations	Definition
PCV	Phishing Composite view
<i>vector</i>	Session Vector
I	Investigator
U	URL
S	SMTP-Email
SS	Multimedia Link
L	Link
WD	WDM
MC	MCAM

The table above illustrates the notations used in the algorithms and Algorithm represents the algorithm to create and manage Machine learning and NLP-based framework as Phishing Detection from Email, Text, and URL systems. In the depicted framework there is a total of 4 stakeholders which I represent Investigator, U for URL, S for SMTP-Email, and L for Link.

This algorithm considers there are U as participants for each stakeholder in the framework consider as User. The Machine learning with NLP and WDM with proper association with MCAM the system depicted the algorithm to

determine the investigative element (URL or Email) text or object is Phishing or is a legitimate text element or object. Respectively where vector = 1 to u. This illustration provides a detailed explanation of how the Investigator interacts for accessing records to determine is Phishing or Legitimate.

The Algorithm Phishing Detection from Email, Text, URL

Procedure System(): Create and update composite view of SMTP, URL and Link Records

Input: SMTP server name, username and password, URL fully qualified domain, Text Records

Output: Identification for each input vector of email, direct URL: is Phishing or is Legitimate

```

1: Fetch email from SMTP server, URL from browser
2: if vector=SMTP.Value && URL.Value then
3: foreach email Ee && url Eu do
4: WDM header Hh = read_extractHead() && object Ho = read_extractObject();
5: if (Hh && Ho indicates that Ee && Eu is Redirective OR HTML encoded) then
6: SAVE: object = event
7: extract and decodedEmail EDd=HTMLDecode(e) && base64_decode(b);
8: parsedEmail pE = emailParser(Hh);
9: headerScore = headerAnalysis(head);
10: linkScore = linkAnalysis(links);
11: textScore = textAnalysis(text);
12: Pcs = combineScore(headerScore, linkScore, textScore);
13: I = SMTP && URL(Y) = {SMTP && URL(Y) | x ∈ Y}
    if i ≥ 1, NLPDetect(SMTP && URL(I))
    Then SI = NLPDetect4 (SMTP && URL(I))
    score(I) = {1+ x(I + a)}/2L
    x = 1
    SA ∪ D and WDM ("url," "link," or "links")
14: Textscore(e) = Max{score(v) | v ∈ e}
15: Objectscore(ev) = maxec ECSimilarity(ev, ec)
16: Final-text-score(e) = 1 if Textscore(e) ≥ 1 and 0 otherwise Objectscore(e) = 1, Final-text-score(e) = 1
17: Finaltext- score(e) = 0 if Textscore(e) < 1 and Final-text-score(e) = 1 otherwise.
18: if Pcs ≥ 2 then
19: Output Label: Phishing
20: end
21: else
22: Output Label: Legitimate
23: end
24: end
25: end
26: end

```

V. NATURAL LANGUAGE PROCESSING FEATURES AND TEST CASES

Some Common Mistakes All the keyword and data preprocessing that we used are listed above. It has then been used for the facilitation and extraction of some unique characteristics. Based on the extracted features' success rates and features, we evaluate the impact of the words with the help of NLP and all available sources to parse the data in the NLP model for classification and detection in various English typewriting styles and techniques.

In the proposed system, we firstly parsed a website address according to the special symbols such as ("=", ".", "/", "&", "?"). Then we characterized the basic list of words and the same is processed to extract distinguishing features which lead us the detect malicious URLs. We identified more than 30 features where we can determine the possibility of phishing pages or URLs. The list of those NLP features is listed in the table below.

- *Test with Word Vectors*

One of many approaches which are used for Machine Learning based text processing is Word Vectorization. The words found in the text are used as features instead of the features in the text-based data that have been drawn out through manual extraction.

The conversion of the words found from the application of the pre-processing process into vectors for all the URLs is shown in Figure 3. The String To Word Vector module in Weka is used for the vectorization process. Machine Learning algorithms are used to process the word vectors that are generated.

TABLE II. NATURAL LANGUAGE PROCESSING TEST FEATURES

The name of feature			
Raw word count (1)	www, com (2)	Alexa check (2)	Known TLD (1)
Punycode (1)	Separated word count (1)	Target keyword count (1)	Average adjacent word length (1)
Average word length (1)	Keyword count (1)	Other words count (1)	Brand check for domain (1)
Longest word length (1)	Brand name count (1)	Digit count (3)	Target brand name count (1)
Shortest word length (1)	Similar keyword count (1)	Subdomain count (1)	Consecutive character repeat (1)
Standard deviation (1)	Special character (8)	Random domain (1)	Similar brand name count (1)
Adjacent word count (1)	Random word count (1)	Length (3)	

- *Hybrid Test*

The Word Vectorization and feature selection steps were repeated for Hybrid Tests. After implementing the feature selection step, more than 200+ word features are acquired. Then the addition of 30+ NLP features to the features obtained was done. Testing of the classification algorithms with more than 250 features has been done.

- *Test Results*

30+, 200+, and 250+ features are used for the tests of NLP, the Word Vectors, and the Hybrid Tests respectively. The values of measuring the obtained achievement values of the tests performed are presented in the table below.

TABLE III. TEST RESULTS

Algorithm	NLP features	Word vector	Hybrid
Random Forest	0.966	0.868	0.972
SMO	0.941	0.865	0.964
Naïve Bayes	0.655	0.817	0.755

From the above table, it is observed that the NLP features system yields better results than the Word Vector features system. If used together as a Hybrid method, NLP Features and Word Vector Features, have a more efficient effect for Random Forest and Sequential Minimal Optimization algorithms.

VI. PERFORMANCE AND TESTING CASES IN MULTI-NODE ENVIRONMENT FOR POSSIBLE DETECTION.

We are using varied Machine Learning and NLP mechanisms and tools to test different scenario-based phishing attempts the details of those are as follows:

We have used 1 attacker virtual machine and multiple client nodes for propagating the phishing page.

Phishing Link creation method:

- Attacker machine: 64Bit Machine with 3.4 Mhz CPU with 16GB RAM
- Attacker Operating system: Kali Linux 2. x
- Web: WebHost services

NLP-equipped Application testing Node for filtering webpages, URLs, E-mail, and SMS (Need to paste SMS contains in application)

- Configuration: 64 Bit Machine running on Virtual Machine
- RAM: 16 GB
- CPU: Multi Core Virtual CPU
- Operating System: Linux
- Platform: Python with Django
- Web: Web Sochekts and Hosting

The algorithm used for testing of Machine learning Model against Test parameters:

- Station Forest (RoF)
- k-Nearest Neighbor (k-NN)

- Support Vector Machine (SVM)
- C4.5 Decision Tree
- Artificial Neural Networks (ANN)
- Random Forest (RF)

TABLE IV. PERFORMANCE RESULTS
Performance of Different ML and NLP Techniques for Detection

	Accuracy			F-Measure			ROC Area		
SVM	96.42	97.28	97.61	0.964	0.972	0.976	0.964	0.989	0.996
k-NN	97.18	97.21	97.23	0.972	0.972	0.972	0.989	0.986	0.985
ANN	96.91	96.94	96.91	0.969	0.969	0.969	0.995	0.982	0.985
Random Forest	97.26	97.28	97.30	0.973	0.973	0.973	0.996	0.991	0.991
CART	95.79	96.98	97.15	0.958	0.97	0.971	0.981	0.996	0.996
C4.5	95.88	97.10	97.24	0.959	0.971	0.972	0.984	0.996	0.997
Rotation Forest	96.79	97.30	97.37	0.968	0.973	0.974	99.40%	0.995	0.997
REP Tree	95.33	96.18	96.95	0.953	0.962	0.97	0.985	0.994	0.995
Random Tree	96.37	97.10	96.99	0.964	0.971	0.97	0.976	0.994	0.994

The results displayed in the table identify that RandomForest attained the finest accuracy of 97.26%.

TABLE V. COMPLEXITY RESULTS
Computational Complexity

	Complexity (sec)		
SVM	104.5	6302.85	8193.72
k-NN	0.02	437.2	349.28
ANN	109.3	550.74	412.59
Random Forest	2.13	25.32	16.77
CART	7.7	48.72	63.31
C4.5	0.44	1.44	2.08
Rotation Forest	11.39	211.58	236.34
REPTree	0.33	1.56	1.4
RandomTree	0.03	0.59	0.39

The above table indicates the Computational Complexity of Different Machine Learning Techniques mainly used for Phishing Attack Detection. As this table clearly showed that the Decision Tree algorithms have low computational complexity. The ANN and SVM attain the highest computational complexity.

VII. COMPARISON OF THIS RESEARCHED APPROACH WITH EXISTING METHODS

The following tables give a clear idea about the Performance, Efficiency, and Throughput of this system in comparison with existing systems by research.

TABLE VI. COMPARISON WITH EXISTING METHODS BASED ON EFFICIENCY AND THROUGHPUT

Sr No	Methods	Precision	Recall	F1 - Score
1	This Method using NLP and ML	97.96	98.96	98.89
2	PhiDMA	91.02	90.55	91.03
3	PPD	89.77	92.05	92.7
4	CPLF	94.89	96.02	95.98

TABLE VII. RUNTIME PERFORMANCE

Sr No	Methods	Runtime (ms)	Recall	F1 - Score
1	This Method using NLP and ML	90	98.96	98.89
2	PhiDMA	2280	90.55	91.03
3	PPD	1893	92.05	92.7
4	CPLF	1089	96.02	95.98

TABLE VIII. RUNTIME ACCURACY OF DETECTION

Sr No	Experiment					
1	SVM dataset (phishing)					
2	NB dataset (phishing)					
3	SVM and NB legitimate dataset (cross-Phishing validation)					
S/N	Work	Phishing data	Benign data	Total	TPR	FPR
1	This Method using NLP and ML	667	1546	3675	99.53	0.45
2	PhiDMA	1078	846	1924	99.44	0.56
3	PPD	4150	4116	8266	99.19	0.81
4	CPLF	2129	995	1662	90.54	5.82

VIII. SUMMARY

In this proposed work, we are developing a system backed with Machine Learning and Natural Language Processing to detect, analyze and further sort out the possible phishing attack from any invited source e.g., E-mail, text message, or any social media platform. This system is equipped with the latest and most effective Machine Learning approaches and using these approaches, we developed a system to cater to various areas to detect possible phishing attacks.

The entire system is developed to identify URL-based, E-mail-body based as well as text message-based Phishing attacks and URLs. The extracted features are then evaluated and form the groups. One is an attribute that tends to be malicious URLs and distinguishes the URLs from legitimate URLs.

Another group is focused on word parsing and using The Random Word Detection Module (RWDM). It applies the vectorization process. Natural Language Processing (NLP) based features used Word Vectors and a Hybrid approach. Alongside Machine Learning Random Forest, Sequential Minimal Optimization, and Naive Bayes algorithms are also in use, which are respectively kernel-based and statistical-based learning models.

On the other hand, we used Maliciousness Analysis Module for identifying malicious content URLs and keywords to determine whether the detected string over keyword is used for any malicious purposes or traps phishing attacks. Overall, we observed a 97% of success rate in the dry run in the simulated environment with parsing just a proposed algorithm. Further study and research will concentrate on designing web-based and implementable systems for all types of users. The same will be incorporated into the project for the next phase. Maintaining 97% accuracy in the detection and analysis phase is an achievement of the fine-tuned NLP algorithm used and backed with CNN as well as Machine Learning.

REFERENCES

- [1] Abdul Basit1 • Maham Zafar1 • Xuan Liu2 A comprehensive survey of AI-enabled phishing attacks detection Techniques Published online: 23 October 2020
- [2] A.Subasi, E. Molah, F. Almkallawi, and T. J. Chaudhery, "Intelligent phishing website detection using random forest classifier," presented at 2017
- [3] A.Subasi et al., "Sensor-Based Human Activity Recognition Using Adaboost Ensemble Classifier," *Procedia Comput. Sci.*, vol. 140, pp. 104–111, 2018.
- [4] Le, A. Markopoulou, and M. Faloutsos, "Phishdef: Url names say it all," presented at the 2011 Proceedings IEEE INFOCOM, 2011, pp. 191–195.

- [5] B.B. Gupta, N. A. Arachchilage, and K. E. Psannis, "Defending against phishing attacks: taxonomy of methods, current issues, and future directions," *Telecommun. Syst.*, vol. 67, no. 2, pp. 247–267, 2018.
- [6] C.L. Tan, K. L. Chiew, and K. Wong, "PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder," *Decis.*
- [7] Detecting Phishing Attacks Using Natural Language Processing and Machine Learning (12 April 2018)
- [8] A. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Mach. Learn.*, vol. 36, no. 1, pp.
- [9] E. Alickovic and A. Subasi, "Ensemble SVM method for automatic sleep stage classification," *IEEE Trans Instrum. Meas.*, 2018.
- [10] F. Feng, Q. Zhou, Z. Shen, X. Yang, L. Han, and J. Wang, "The application of a novel neural network in the detection of phishing websites," *J. Ambient Intell.*
- [11] Feature Selections for the Classification of Webpages to Detect Phishing Attacks: A Survey
- [12] G. Ramesh, I. Krishnamurthi, and K. S. S. Kumar, "An efficacious method for detecting phishing webpages through target domain identification," *Decis.*
- [13] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "Cantina+: A feature-rich machine learning framework for detecting phishing websites," *ACM Trans. Inf. Syst.*
- [14] Heuristic Phishing Detection and URL Checking Methodology Based on Scraping and Web Crawling 2020 Romulo F. Almeida, C. Westphall Published 2020
- [15] <https://github.com/zrz1996/Spam-Email-Classifier-DataSet.git>
- [16] <https://ieeexplore.ieee.org/document/8093406>
- [17] https://link.springer.com/content/pdf/10.1007%2F978-3-642-33167-1_47.pdf
- [18] <https://monkey.org/~jose/phishing/>
- [19] <https://research.aalto.fi/en/datasets/phishstorm-phishing-legitimate-URL-dataset>
- [20] <https://towardsdatascience.com/phishing-domain-detection-with-ml-5be9c99293e5>
- [21] <https://www.kaggle.com/akashkr/phishing-website-dataset>
- [22] *Humans. Comput.*, pp. 1–15, 2018.
- [23] *Imaging*, vol. 29, no. 2, pp. 531–542, 2010.
- [24] *Intelligent Systems Design and Applications*, Springer Science and Business Media LLC, 2018
- [25] International Conference on Electrical and Computing Technologies and Applications (ICECTA), 2017, pp. 1–5.
- [26] International Conference on Intelligent Systems Design and Applications ISDA 2017
- [27] International Conference on Semantic Computing (ICSC), 2018, pp. 300–301.
- [28] International Conference on World Wide Web, 2007, pp. 639–648.
- [29] L. I. Kuncheva, J. J. Rodríguez, C. O. Plumpton, D. E. Linden, and S. Johnston, "Random subspace ensembles for fMRI classification," *IEEE Trans. Med.*
- [30] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent phishing detection system for e-banking using fuzzy data mining," *Expert Syst. Appl.*, vol.
- [31] M. Babagoli, M. P. Aghababa, and V. Solouk, "Heuristic nonlinear regression strategy for detecting phishing websites," *Soft Comput.*, vol. 23, no. 12, pp.
- [32] M. He et al., "An efficient phishing webpage detector," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 12018–12027, Sep. 2011.
- [33] M. Yaman, A. Subasi, and F. Rattay, "Comparison of Random Subspace and Voting Ensemble Machine Learning Methods for Face Recognition," *Symmetry*,
- [34] Machine learning-based phishing detection from URLs 2019. K.
- [35] Sahingoz, Ebubekir Buber, +1 author B. Diri Published 2019
- [36] P. A. Barraclough, M. A. Hossain, M. A. Tahir, G. Sexton, and N. Aslam, "Intelligent phishing detection and protection scheme for online transactions,"
- [37] PhishDump: A multi-model ensemble-based technique for the detection of phishing sites in mobile devices R. S. Rao, T. Vaishnavi, R. Pais Published 2019
- [38] R. Islam and J. Abawajy, "A multi-tier phishing detection and filtering approach," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 324–335, 2013.
- [39] R. S. Rao and A. R. Pais, "Jail-Phish: An improved search engine based phishing detection system," *Comput. Secur.*, vol. 83, pp. 246–267, 2019.
- [40] S. C. Jeeva and E. B. Rajasingh, "Intelligent phishing URL detection using association rule mining," *Hum.-Centric Comput. Inf. Sci.*, vol. 6, no. 1, p. 10, 2016.
- [41] S. Gastellier-Prevost, G. G. Granadillo, and M. Laurent, "Decisive heuristics to differentiate legitimate from phishing sites," presented at the Network and Information Systems Security (SAR-SSI), 2011 Conference on, 2011, pp. 1–9.
- [42] S. R. Curtis, P. Rajivan, D. N. Jones, and C. Gonzalez, "Phishing attempts among the dark triad: Patterns of attack and vulnerability," *Comput. Hum. Behav.*, vol. 87, pp. 174–182, 2018.
- [43] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decis.*
- [44] S. Smadi, N. Aslam, and L. Zhang, "Detection of online phishing email using dynamic evolving neural network based on reinforcement learning," *Decis.*

- [45] Secure. TISSEC, vol. 14, no. 2, p. 21, 2011.
- [46] Support Syst., vol. 107, pp. 88–102, 2018.
- [47] Support Syst., vol. 107, pp. 88–102, 2018.
- [46] Support Syst., vol. 61, pp. 12–22, May 2014.
- [47] Support Syst., vol. 88, pp. 18–27, 2016.
- [48] T. Peng, I. Harris, and Y. Sawa, “Detecting phishing attacks using natural language processing and machine learning,” presented at the 2018 IEEE 12th
- [49] vol. 10, no. 11, p. 651, 2018.
- [50] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, “A stacking model using URL and HTML features for phishing webpage detection,” *Future Gener. Comput. Syst.*, vol. 94, pp. 27–39, 2019.
- [51] Y. Zhang, J. I. Hong, and L. F. Cranor, “Cantina: a content-based approach to detecting phishing websites,” presented at the Proceedings of the 16th