EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Learning to Detect Phishing Web Pages Using Lexical and String Complexity Analysis

Dharmaraj Patil[1], Tareek Pattewar[2,*], Shailendra Pardeshi[1], Vipul Punjabi[1] and Rajnikant Wagh[1]

[1]Department of Computer Engineering, SES's RC Patel Institute of Technology, Shirpur, India
[2]Department of Computer Engineering, Vishwakarma University Pune, India

Abstract

Phishing is the most common and effective sort of attack employed by cybercriminals to deceive and steal sensitive information from innocent Web users. Researchers have developed major solutions to deal with this problem in recent years, but there are still a number of open challenges due to the ever-changing nature of phishing attacks. To discriminate between benign and phishing URLs, this paper proposes a static method based on lexical and string complexity analysis and distinguishing URL features. Proposed approach has been evaluated on the basis of two state of the art online learning classifiers. The confidence weighted learning classifier achieved a significant phishing URL detection accuracy of 98.35 %, error-rate of 1.65%, FPR of 0.026 and FNR of 0.005. Also, adaptive regularization of weight classifier achieved accuracy of 97.28%, error-rate of 2.72%, FPR of 0.000 and FNR of 0.052. Similar approach shows the improvement in the detection of the phishing web pages.

Keywords: Phishing detection, Lexical analysis, Entropy, Kolmogorov complexity, Huffman coding complexity, online machine learning, cyber security

Received on 22 February 2022, accepted on 19 April 2022, published on 20 April 2022

doi: 10.4108/eai.20-4-2022.173950

*Corresponding author. Email: tareek.pattewar@vupune.ac.in

1. Introduction

Phishing is a type of fraud that involves utilizing e-mail, social engineering, and technical deception to acquire and misuse a victim's personal information, such as usernames, passwords, credit card numbers, account numbers, and so on. The number of phishing websites appears to be growing at an alarming rate. A variety of attacks are launched with the goal of convincing Web users that they are communicating with a trusted entity. According to the most recent APWG Phishing Activity Trends Report, Phishing Activity, 2Q 2020 - 1Q 2021, phishing is still at an all-time high: January 2021 breaks all records [1],

- The APWG set a new record in January 2021, with 245,771 attacks in a single month.
- Business e-mail compromise scams are becoming more expensive, with average wire transfer requests in BEC attacks increasing to $85000, up from $48000 in the third quarter of 2020.
- During the holiday shopping season, Business Email Compromise (BEC) attacks used gift cards to cash out.
- In this quarter, the financial institution, webmail, and social media sectors were the most frequently targeted by phishing.
- After steadily increasing for years, the use of HTTPS encryption on phishing sites has plateaued at 83%.
- Phishers continue to obtain domain names for their schemes through specific domain name registrars.

According to the recent PhishLabs Threat Trends and Intelligence Report Q1 2021, following are the key points [2],

- Phishing is on the rise, with phishing sites discovered in Q1 2021 outnumbering those discovered in Q1 2020 by 47%.
- Sixty-two percent of all phishing sites made use of free online services and tools.
- Malware Delivery ZLoader was responsible for 62% of email-based payloads aimed at corporate users.
- Although SSL certificates were used in 82.7% of phishing attacks, Q1 was the first quarter in which there was no significant increase in SSL usage.
- 44.5% of all credential theft phishing emails reported by corporate users targeted Office 365 accounts.

Due to the pandemic in 2021, the phishing environment has changed in the previous year 2020, as there have been drastic changes in daily life. In order to land their new scams, phishers and attackers have attempted to take advantage of the various working environments and new tools being used for work from home. The year 2020 saw a record increase in phishing sites, with Google detecting 2.11m phishing sites, a 25% increase over 2019. It appears that cyber criminals will increase their efforts in 2021, with 64% of businesses expecting an increase in COVID'19 related phishing emails in 2021. Phishing assaults are on the rise, according to the two Security Threat reports mentioned above, wreaking havoc on businesses, banks, social networks, and unsuspecting consumers. Researchers have developed substantial solutions to cope with this challenge in recent years, but because to the ever-changing nature of phishing assaults and cyber-criminals' inventive out-of-the-box thinking, there are still numerous outstanding concerns [3].

Following are some of the phishing detection challenges,

- Cybercriminals understand that it takes time for relevant data about domains, URLs, sources, and any highlighted things to be acquired, reviewed, and banned. They take advantage of this by shortening the lifespan of bogus sites by swapping domains and URLs, typically within hours or even minutes.
- Cybersecurity systems have typically relied on patches for things like malware and blacklists; however, IT workers are sometimes hesitant to deploy fixes, especially if they are required frequently.
- An enormous number in a relatively short period of time, and these are only the ones that were discovered. This takes us back to the significance of speed and real-time scanning. A phishing detection system that can verify an email or link while the user is reading it and finish its check-in in fractions of seconds is crucial.
- The fact that the assaults are both targeted and multi-channel is one of the most difficult difficulties in the anti-phishing sector. Previously, these sorts of frauds could only be perpetrated over email and on PCs. Any sort of device, from PCs to smartphones, and every communication medium, from emails to social media and even voicemails, may now be used to launch an assault.
- Bad actors exploit trustworthy technology that we all use in our daily lives to make their communication appear more real. Often, a corporate email server will be attacked if an employee unwittingly installs malware or spyware after opening a file from a Google Drive or Dropbox link.

The main focus of this paper is to detect the short-lived phishing Webpages.

This paper proposed a static method for detecting phishing URLs based on lexical and string complexity analysis of the URL string. We used 15 different static URL features. A balanced binary labeled dataset of 350404 phishing and benign URLs is created. It has 175202 benign URLs and 175202 phishing URLs. To assess our approach, we used two cutting-edge supervised online learning classifiers: Adaptive Regularization of Weights (ARW) and Confidence Weighted Learning (CW) (AROW). Experiments on our binary dataset are conducted using the aforementioned online machine learning classifiers. The CW classifier was shown to have a significant phishing URL detection accuracy of 98.35 percent, a 1.65 percent error rate, an FPR of 0.026, and a FNR of 0.005. Also, AROW classifier achieved accuracy of 97.28%, error-rate of 2.72%, FPR of 0.000 and FNR of 0.052. The major contributions of this paper are as follows:

- We proposed an entropy and string complexity (Kolmogorov and Huffman coding complexity) based approach for detecting phishing URLs, and our experimental results are promising.
- It was discovered that combining string complexity features with lexical analysis of URLs improved the detection performance of phishing URLs significantly.
- When compared to the dynamic Web page feature extraction approach, our approach is based on static analysis of URL strings, which significantly reduces the time required for feature collection, training, and testing of classifier models.
- The proposed approach is evaluated using two cutting-edge online machine learning classifiers (CW and AROW) which achieved significant improvement in the detection performance of phishing URLs with minimum overhead.

The rest of this paper is structured as follows. Section 2 provides a brief related work. The methodology is described in Section 3 using feature extraction and supervised online machine learning algorithms. The results of the experiments are presented in Section 4. In Section 5, the discussion is given. Finally, in Section 6, the conclusion is presented.

2. Related Work

Several approaches for detecting phishing URLs have been proposed. In this section, we will provide a brief overview of a few cutting-edge approaches.

Zabihimayvan et al. investigated a consensus on the definitive features to be used in phishing detection. To select the most effective features from three benchmark data sets, they used Fuzzy Rough Set (FRS) theory as a tool. Three widely used phishing detection classifiers are given the relevant features. To test the FRS feature selection, the classifiers are trained on a different out-of-sample data set of 14000 website samples. When Random Forest classification is applied, the maximum F-measure attained by FRS feature selection is 95 percent, according to their testing data. FRS also selected nine universal features from all three data sets. The F-measure value using this universal feature set is approximately 93% [4].

For detecting phishing website obfuscation tactics and enhancing the filtering efficiency of authentic web pages, Ding et al. suggested the Search and Heuristic Rule and Logistic Regression (SHLR) combination detection method. There are three steps to the technique. The webpage's title tag content is first submitted as search keywords into the Baidu search engine, and if the domain name of any of the top-10 search results appears, the webpage is regarded legitimate; otherwise, additional review is undertaken. Second, if the webpage cannot be identified as legitimate, the heuristic rules given by the character features are used to decide whether it is a phishing page. The first two processes can filter web pages quickly to fulfil real-time detection needs. Finally, the remaining pages are evaluated using a logistic regression classifier in order to increase the detection method's adaptability and accuracy. According to their test results, the SHLR can filter 61.9 percent of authentic URLs and identify 22.9 percent of phishing web pages based on URL lexical information.
The SHLR is 98.9% accurate, according to them [5].

Niakanlahiji et al. proposed PhishMon, a new features rich machine learning frameworks for detecting phishing web pages. According to them, it is built a collection of fifteen new properties that could be computed efficient way from webpages without the usage of third-party service like search engines or “WHOIS” servers. These qualities capture many aspects of lawful web apps and the web infrastructures that support them. For phishers, emulating these components is costly because it necessitates significantly require more time and efforts on their underlying infrastructures and web apps, in addition to the time and effort required to replicate appearance of targeted websites. They show that PhishMon can detect undetected phishing from regular web sites with the high degree of accuracy after extensive testing on a datasets containing 4800 distinct phishing and 17500 separated benign online pages. PhishMon achieved 95.4 percent accuracy with a 1.3 percent false positive rate (FPR) on a dataset comprising unique phishing incidents [6].

Yuan et al. proposed extracting features from URLs and webpage links to detect phishing websites and their targets. The fundamental features of the links on the supplied URL's webpage, as well as the basic properties of the given URL, such as length, suspicious characters, and the amount of dots, are used to create a feature matrix. Each column of the feature matrix is also used to extract statistical features such as mean, median, and variance. The given URL, links, and information on its webpage are also used to extract lexical properties such as title and textual content. A number of machine learning models for phishing detection have been examined, according to them, with the Deep Forest model exhibiting competitive performance, with a true positive rate of 98.3% and a false alarm rate of 2.6 percent. They devised a successful phishing target detection approach based on search operators via search engines, with an accuracy of 93.98 percent [7].

Babagoli et al. suggested a method for detecting phishing websites that combines a feature selection approach with a meta-heuristic-based nonlinear regression algorithm. To test the suggested strategy, they employed a dataset of 11055 phishing and authentic online pages, and they chose 20 features to extract from the websites. They employed two feature selection approaches to find the optimum feature subset: decision tree and wrapper, with the wrapper method achieving detection accuracy rates of up to 96.32 percent. They employed two Meta heuristic algorithms, such as harmony search (HS), which used a non-linear regression technique as well as support vector machine, to anticipate and detect bogus websites after the feature selection stage (SVM). According to them, the HS method was used to generate the parameters of the suggested regression model, and the nonlinear regression approach was used to classify the websites. According to the experimental data, the nonlinear regression based on HS obtained accuracy rates of 94.13 percent for the train and 92.80 percent for the test processes. In a performance comparison, the nonlinear regression-based HS surpasses SVM [8].

Arab et al. proposed a new clustering-based detection method for phishing websites. They presented a weighted version of Euclidean distance to improve clustering performance. According to them, using weights to correct the membership of records in clusters has resulted in very good results that are comparable to the results of classification approaches. They used 30 key features of websites to determine whether they were phishing or not. They conducted the experiments on Huddersfield University's dataset. The work's implementation results have been evaluated and compared to other supervised classification methods such as decision trees and artificial neural networks. In terms of accuracy, their experimental results show that the proposed method outperforms other classification and clustering algorithms [9].

Dharmaraj Patil and colleagues suggested a multi-class classification-based methodology for detecting malicious URLs and attack types. In this paper, they presented 42 new useful features for phishing, spam and

URLs malware. Out of 49935 malicious and benign URLs, they created a binary and multi-class dataset. In total, there are 26041 benign URLs and 23894 bad URLs, with 11297 malwares, 8976 phishing, and 3621 spam URLs. To test the suggested method, they used supervised batch as well as online machine learning classifiers. They discovered the confidence weighted machine learning classifiers which achieve best average among the detection accuracy up to 98.44 percent having 1.56% error rates in multi-class labeling as settings and 99.86% detection accuracy with avoidable error rates of 0.14% in the binary settings using their new proposed URL feature [10].

Basnet and colleagues suggest a machine learning based method for detecting phishing web pages. They had used 6 Batch Learning algorithms, Random Forests, Support Vector Machines (SVMs) having rbf linear kernel, Naïve-Bayes, C4.5, Logistic Regression (LR), and a set of 5 online learning algorithms-updatable version of Naïve-Bayes (NBU), updatable version of Logit-Boost (LB-U), Perceptron (MLP), Passive-Aggressive (PA) and Confidence Weighted (CW) algorithms. They employed 179 Web page features to show their technique, including lexical-based features, keyword-based features, search engine-based features, and reputation-based features. The WEKA and CW libraries were used in all of the tests. Their proposed approach accurately detects phishing Web pages with 99.9% accuracy, a 0.00% false positive rate (FPR), and a 0.06 percent false negative rate (FNR) [11].

To detect zero-day phishing attacks, Mishra et al. proposed a novel intelligent phishing detection system, CSS and URI matching-based phishing detection system (CUMP). It is based on the concept of matching uniform resource identifiers (URIs) and cascading style sheets (CSSs). They used basic properties of any phishing attack for URIs and CSSs matching to defend the against phishing website attacks, particularly 'zero-day' attacks. Their proposed solution, according to them, is very effective in the detecting a wide range of the website phishing attacks, with True Positive and True Negative rates of 93.27% and 100% respectively and results in a lower False Positive Rate [12].

By proposing a novel browser architecture, HR, M. G et al. have proposed a novel technique for detecting the phishing website on client sides. They used the rule of extraction framework in this system to extract the properties or features of website using only URLs. List contains 30 different URL properties that will be used later by the Random Forest Classification machine learning model to determine the authenticity of the website. To train the model, they used a dataset with 11055 records. They developed this technique to ensure maximum securities and 99.36% accuracy in detecting phishing websites in real-time [13].

Adebowale et al. presented an Adaptive Neuro-Fuzzy Inference System (ANFIS)-based robust scheme for web phishing detections and protections based on the integrated feature of Text, Images, and Frames. The proposed solution, according to them, achieves 98.3% accuracy [14].

Cooper et al. proposed the use of audiovisual alerts and warnings to reduce phishing susceptibility on mobile devices. This study, according to them, is divided into three phases. During the first phase, 32 subject matter experts provided feedback on a phishing alert and warning system. The second phase included the creation of a phishing alert and warning system prototype as well as a pilot study to validate it. The Phishing Alert and Warning System was distributed to 205 participants during the third phase. According to experimental results, audio and visual warnings in emails may reduce phishing susceptibility [15].

RAIDER: Reinforcement AIded Spear Phishing DEtectoR was created by Evans et al. It is a feature evaluation system based on reinforcement learning that can automatically find the best features for detecting various types of attacks. Experiment results from RAIDER over 11000 emails and across three attack scenarios suggest that using reinforcement learning to automatically identify the significant features could reduce the required feature dimensions by 55% when compared to existing ML-based systems. According to them, it has increased the accuracy of detecting spoofing attacks by 4%, from 90% to 94% [16].

Mohammad et al. proposed an artificial neural network-based anti-phishing model for enterprises. This model, according to them, effectively determines whether the phishing email is known phishing or unknown phishing. They used the Feed-Forward Back propagation and Levenberg-Marquart methods of Artificial Neural Network (ANN) to improve the URL classification process, as well as the Fuzzy Inference System to obtain results with imprecise social feature data [17].

Jain et al. offered a comprehensive review of phishing assaults, their exploitation, and some of the most recent visual similarity-based phishing detection and comparison algorithms. They used visual similarity-based phishing detection approaches including text content, text format, HTML elements, Cascading Style Sheet (CSS), image, and so on to make their judgement [18].

To detect and regulate the phishing problem, Mourtaji et al. developed new hybrid rule based solution that integrates six different algorithmic models. According to them, black listed technique, the lexical and host method, the content method, the identity method, the identity similarity method, the visual similarity method, and the behavioral approach are among the 37 attributes retrieved from six different ways. They compared Machine Learning and Deep Learning models including CART (Decision Trees), SVM (Support Vector Machines), and KNN (K-Nearest Neighbors), as well as deep learning models like MLP (Multilayer Perceptron) and CNN (Convolutional Neural Networks). They had the best deep learning accuracy, with scores of 97.945 for CNN Model and 93.216 for MLP Model [19].

Akdemir et al. examined the content of 208 coronavirus-themed phishing emails. They discovered

nine different types of phishing messages created by phishers. According to them, they analyzed coronavirus-themed phishing emails and discovered a shift in phishers' tactics [20].

El Aassal et al. presented a new feature taxonomy on interpretation and function of each feature. They've also proposed the 'PhishBench' benchmarking framework, which allows for a systematic and thorough evaluation and comparison of existing feature for phishing detections under identical all experimental conditions, including a unified systems specification, dataset, classifier and evaluation metrics [21].

Sahingoz et al. proposed a real-time anti-phishing system based on NLP-based features and seven different categorization algorithms. According to them, the system's qualities include language independence, the use of a huge amount of phishing and legitimate data, real-time execution, detection of new websites, independence from third-party services, and the use of feature-rich classifiers. According to the experimental and comparison findings of the built classification methods, the Random Forest algorithm with only NLP-based characteristics outperforms the others, detecting phishing URLs with a 97.98 percent accuracy rate [22].

Butnaru et al. used supervised machine learnings to detect and prevent phishing attacks based on Novel Combination of feature extracted solely from URLs. They compared the system's performance over time with a dataset of active phishing attacks to Google safe browsing (GSB), which is the default securities control in most demanding web browsers. According to them, their work outperformed GSB in their experiments and performed well bitterly against phishing URL that are still active a year after their model was trained [23].

Bagui et al. proposed a novel solution based on deep semantic analysis to capture inherent text body characteristics. They used one-hot encoding in conjunction with DL and ML techniques to classify emails as phishing or not. They used ML models such as Nave Bayes, SVM, and Decision Trees, as well as DL models such as Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM). According to them, DL models outperformed ML models in terms of accuracy, but ML models outperformed DL models in terms of computation time. They achieved the highest accuracy (96.34%) using CNN with Word Embedding and demonstrated the effectiveness of semantic analysis in detecting phishing emails [24].

Zhu et al. introduced OFS-NN, an effective phishing website detection model based on an optimal feature selection approach and a neural network. They first develop a new metric, feature validity value (FVV), to assess the influence of sensitive features on phishing website detection, according to them. They devised an algorithm based on the new FVV index to identify the top features from phishing websites. They trained the underlying neural network using the best features available, and then developed an optimal classifier to detect phishing websites. The OFS-NN model is accurate and stable in detecting a variety of phishing websites, according to the results [25].

To improve phishing website identification, Ali et al. suggested an intelligent phishing website detection approach based on particle swarm optimization-based feature weighting. According to them, the proposed method involves employing particle swarm optimization (PSO) to efficiently weight numerous website attributes in order to detect phishing websites with more accuracy. The proposed PSO-based feature weighting increased the classification accuracy, true positive and negative rates, and false positive and negative rates of machine learning models while employing less website features [26].

Mao et al. proposed a learning-based aggregation analysis mechanism for determining page layout similarity, which can be used to detect phishing pages. They hoped to use machine learning techniques to enable automated page-layout-based phishing detection techniques. They prototyped their solution and assessed the accuracy and factors influencing their results of four popular machine learning classifiers [27]. Acharya et al. created PhishPrint, a novel, scalable, low-cost framework for evaluating web security crawlers against multiple cloaking attacks [28].

CyberPulse++, a machine learning-based security system described by Rasool, R. U., et al., uses a pre-trained machine-learning repository to evaluate collected network statistics in real-time to detect abnormal route performance on network links. It efficiently addresses various issues faced by network security solutions, according to them, including the feasibility of large-scale network-level monitoring and data collecting. They have shown that the system can proactively identify and fight against link flooding attacks in real time with little bandwidth and computational overhead [29].

Vimalachandran, P. et al. have reviewed and addressed the impact of data security and privacy on the use of the MyHR system and its associated issues. They have determined and analyzed where privacy becomes an issue of using the MyHR system. Also, they have presented an appropriate method to protect the security and privacy of the MyHR system in Australia [30].

3. Methods and Materials

3.1. Proposed framework of detecting phishing URLs

Figure 1 depicts the framework of our proposed phishing URL detection system. Our methodology consists of feature extraction, pre-processing, and labeling, as well as training and detection using supervised online machine learning algorithms.

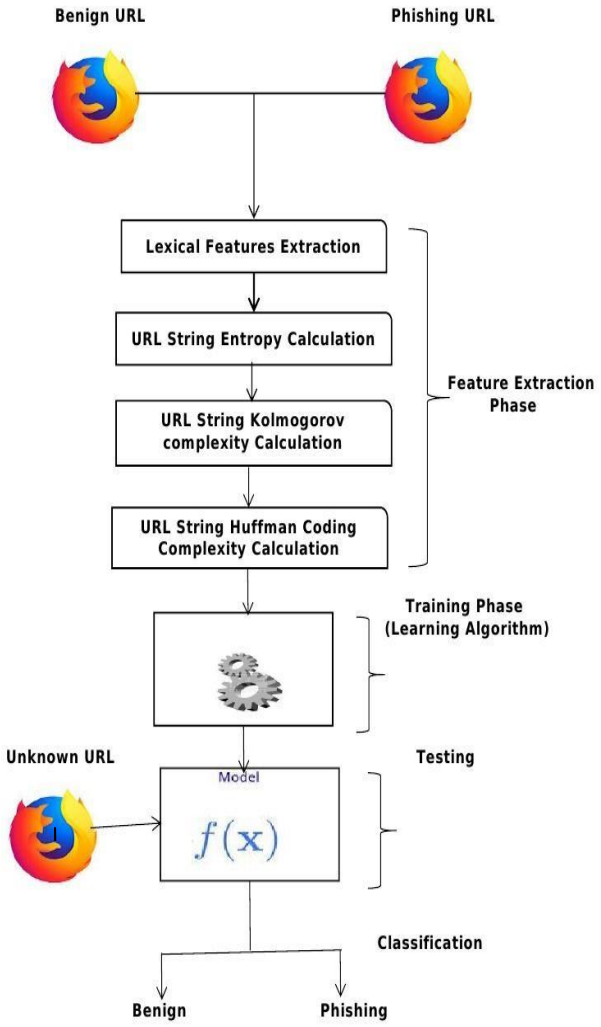

Figure 1. Architecture of our proposed phishing detection system for URLs

3.2. Extraction of Features, pre-processing techniques and labeling

The Java-based feature extraction module receives phishing and benign URLs from benchmarks sources. Table 1 shows the 15 static features extracted from benign and phishing URLs. These are both numerical and physical characteristics. Phishing URLs are labeled as +1 in binary dataset preparation, while benign URLs are labeled as -1. We extracted four kinds of static URL features: lexical features, URL String Entropy, URL String Kolmogorov complexity, and URL String Huffman Coding Complexity. We constructed a URL feature extractor in Java. The features are acquired through lexical scanning of the URL string [31,32], and the URL feature extraction is accomplished using Java's URL class.

3.3. Lexical features

These characteristics are derived from the URL name or the URL string. These are the "look and feel" properties of the URL string that can be used to determine the phishing nature of a URL. The most commonly used lexical features are statistical properties of the URL string, such as URL length, number of special characters, and so on. The URL string contained 12 lexical features that we extracted. We examined the URL string for suspicious lexical characters such as =,. -, /, $,, =,?, percent, &, and. The reason for choosing these features is that generally benign URLs do not contain such special characters. Benign URLs typically include. / and? special characters, for example, https://www.alexa.com/siteinfo/facebook.com. The phishing URL, https://4k.smarttv-magazine-luiza.dns-cloud.net/tv-4kuhd/smart-tv-4k-led-60-lg60uk6200-wi-fi-hdr-inteligencia-artificial-conversor-digital-3hdmi.php?ass=NwDu!Z1*, contains most of these special characters to mimic some benign URL.

Table 1. Static features of the benign and phishing URLs.

Sr No	Feature Name	Type
1	URL Length	numeric
2	No. of Tokens in URL	numeric
3	No. of (.) Symbols in URL	numeric
4	No. of (-) Symbols in URL	numeric
5	No. of ( ) Symbols in URL	numeric
6	No. of (=) Symbols in URL	numeric
7	No. of (/) Symbols in URL	numeric
8	No. of (%) Symbols in URL	numeric
9	No. of (?) Symbols in URL	numeric
10	No. of (&) Symbols in URL	numeric
11	No. of (@) Symbols in URL	numeric
12	No. of ($) Symbols in URL	numeric
13	Entropy of URL string	real
14	Kolmogorov Complexity of URL	real
15	Huffman Complexity of URL	real

Table 2 and Figure 2 shows the average frequency (AF) of occurrence of these special symbols in the benign and phishing URLs. Suppose N is a set of phishing and benign URLs and X = x1, x2...., xn is the frequency of

occurrence of special symbols, then AF is given by equation (1),

$$AF = \frac{\sum_{i=1}^{N} x_i}{N} \quad (1)$$

where,
AF = Average frequency of special symbol occurrence in URL
x_i = i^{th} Feature value
N = Total number of URLs.

Attackers utilise a variety of obfuscation techniques to imitate the names of benign URLs in order to "appear" like them. As demonstrated in Table 2, the average frequency of occurrence in phishing URLs is higher than in benign URLs for the majority of the special symbols attributes. As a result, these characteristics are useful in distinguishing phishing URLs from benign URLs.

Table 2. Average frequency occurrence of special symbols in benign and phishing URLs in our dataset

Feature Name	Benign URL	Phishing URL
Length of URL	24.20	86.77
No. of Tokens in URL	8.42	21.46
No. of Dots (.)	2.12	2.71
No. of Hyphens (-)	0.09	0.82
No. of Underscore ( )	0.00	0.37
No. of Equal (=)	0.00	0.65
No. of Fslash (/)	2.00	5.41
No. of mod (%)	0.00	0.15
No. of Question Mark Sign (?)	0.00	0.25
No. of ampersand (&)	0.00	0.39
No. of at the Rate (@)	0.00	0.03
No. of dollar ($)	0.00	0.03

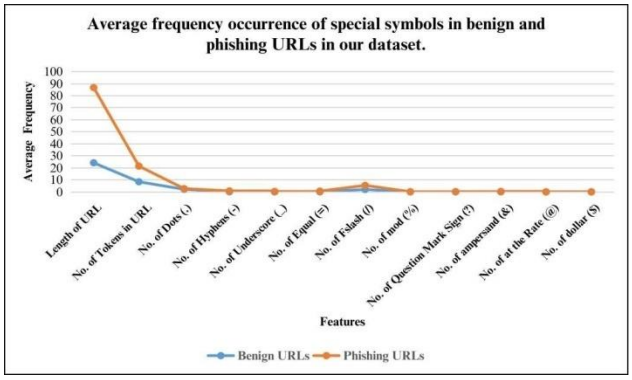

Figure 2. Average frequency occurrence of special symbols in benign and phishing URLs in our dataset

3.4. String complexity features

- Entropy

We used entropy as a measure to highlight the unpredictability component in URLs: the higher the entropy, the higher the randomness of the URL string under evaluation. Each benign and phishing URL's entropy has been calculated separately. The following equation [33] is used to compute the entropy of the URL string.

$$H(x) = -\sum_{i=0}^{n} p(x_i) log_b p(x_i) \quad (2)$$

where,
x - URL string,
H(x) - Entropy of URL string x,
b - Base of the logarithm used and
p(x) – Probability mass function.
Following are the examples of phishing and benign URLs to illustrate the above,
a. Phishing URL: http://pelangsingbsh.com/wp-includes/css/validation./updateboa/BOA
Entropy H(x)= 4.40
b. Benign URL: http://www.linkedin.com
Entropy H(x)= 3.88

In comparison to benign URLs, phishing URLs have high entropy, as shown in Table 3. It demonstrates that phishing URLs have a higher unpredictability factor, indicating that they are phishing.

Table 3. Entropy Average of benign as well as phishing URL used in datasets

Benign URL	Phishing URL
3.74	4.42

- **Kolmogorov complexity**

The Kolmogorov complexity of an object is a measure of its descriptive complexity. It is the smallest program length that a universal computer can produce in order to generate a specific sequence. For a random sequence, the expected value of K(x) is roughly equal to the entropy of the source distribution for the process that generated the sequence, which is related to Shannon entropy. In contrast to entropy, Kolmogorov Complexity is concerned with the string in question rather than the source distribution [34,35].

Random strings have a high Kolmogorov Complexity on the order of their length since patterns cannot be identified to minimize the size of a

program that generates such a string. Strings with a lot of structure, on the other hand, are quite simple. We concentrate our efforts on the URL string, which includes, among other things, the domain name, path, filename, and query. We want to identify whether a URL is phishing or not using the Kolmogorov complexity measure. Because their patterns cannot be detected to replicate the true URL, phishing URL strings are more random, according to our observations, resulting in a high K(x). The benign URL string, on the other hand, has a lower Kolmogorov Complexity K(x).

Following are the examples of phishing and benign URLs to illustrate the above,

a. Phishing URL:
http://pelangsingbsh.com/wp-includes/css/validation./updateboa/BOA
Kolmogorov Complexity K(x)= 44.00
b. Benign URL:
http://www.linkedin.com
Kolmogorov Complexity K(x)= 19.00

Table 4 shows the average Kolmogorov Complexity K(x) of the benign and phishing URLs in our dataset. It is clear from Table 4, that phishing URLs are more random in nature than benign URLs resulting in high Kolmogorov Complexity.

Table 4. Average Kolmogorov Complexity K(x) of the benign and phishing URL in our datasets

Benign URL	Phishing URL
18.64	53.50

- **Huffman Coding Complexity**

Huffman coding is a method developed by David Huffman for discovering the lowest-cost prefix-free codes. A collection of typically strictly positive symbol weights, as well as an alphabet of n different symbols indicated by the numbers 0 to n -1, is assumed to be provided [36].

Some URLs, whether phishing or not, may have common patterns, which is relevant to our work. Huffman's method employs a frequency table for each symbol (or character) in the input. Then we must assign each character a variable-length bit string that unambiguously reflects that character. This means that each character's encoding must have a distinct prefix. The Huffman coding compression ratio (CR) for URL string x is as follows,

$$CR(x) = \frac{Original\_string\_length}{Encoded\_string\_length} \quad (3)$$

Duplicated characters were discovered in URLs, according to our findings. Phishing URLs have greater redundancy in characters than benign URLs because phishers typically use obfuscation to mimic benign URLs. Huffman's method uses a table of symbol (or character) frequency of occurrence in the input URLs string, then assigns a variable-length bit string to each character. This indicates that a greater occurrence frequency results in a shorter bit string and a higher compression ratio. As a consequence, compressing the URL strings with Huffman compression offers a good indication of the complexity. Here are some phishing and benign URL samples that show the aforementioned.

a. Phishing URL:
https://interacposcentre-ca.serveirc.com/Interac/banks/CIBC/accountConfirm.php
Huffman Compression Ratio CR(x) = 0.86
b. Benign URL:
http://www.google.com
Huffman Compression Ratio CR(x)= 1.14

Table 5 shows the average Huffman Compression Ratio CR(x) of the benign and phishing URLs in our dataset.

Table 5. Huffman Compression Ratio *CR*(*x*) of the benign and phishing URLs in our datasets

Benign URL	Phishing URL
1.09	0.86

3.5. Feature Representation

We employed classification techniques, which necessitated the usage of sparse vectors to store URL characteristics. In machine learning, a lot of data is sparse, meaning it's primarily zeros and binary. In Java, we developed a feature pre-processing module that removes any zero-value features from the dataset. The smaller feature subset improved the training time noticeably as a result of this strategy. Our feature vector for the binary URL dataset is shown in Figure 3.

```
+1 1:86 2:30 3:3 6:1 7:9 9:1 11:1 13:4.51 14:53.00 15:0.83
-1 1:28 2:8 3:2 7:2 13:4.11 14:22.00 15:1.03
+1 1:87 2:31 3:3 6:1 7:9 9:1 10:1 11:1 13:4.55 14:54.00 15:0.82
-1 1:22 2:8 3:2 7:2 13:3.75 14:18.00 15:1.13
+1 1:51 2:16 3:2 7:6 13:4.24 14:33.00 15:0.94
-1 1:33 2:8 3:2 7:2 13:3.92 14:24.00 15:1.01
+1 1:58 2:20 3:2 4:2 7:6 13:4.28 14:36.00 15:0.90
-1 1:24 2:8 3:2 7:2 13:3.54 14:17.00 15:1.08
+1 1:88 2:26 3:6 7:7 13:3.97 14:47.00 15:0.88
-1 1:16 2:8 3:2 7:2 13:3.03 14:12.00 15:1.20
+1 1:111 2:34 3:7 6:1 7:7 9:1 11:1 13:4.17 14:59.00 15:0.84
-1 1:24 2:8 3:2 7:2 13:3.64 14:18.00 15:1.13
+1 1:114 2:34 3:7 6:1 7:7 9:1 11:1 13:4.26 14:63.00 15:0.83
-1 1:23 2:8 3:2 7:2 13:3.68 14:17.00 15:1.10
+1 1:206 2:75 3:8 4:1 6:7 7:7 8:1 9:1 10:6 11:1 13:4.88 14:110.00 15:0.68
```

Figure 3. Feature vector for binary URL dataset

3.6. Machine Learning Algorithms for Phishing URLs Detection

We have used online learning algorithms like confidence weighted (CW) learning and adaptive regularization of weight vectors (AROW) for phishing URLs detection. Online learning algorithms are fast, simple, make few statistical assumptions and perform well in a wide variety of settings. The algorithms in the internet world work in rounds. In round I an online algorithm receive xi and, given the current model, predicts xi's label as y_i. The true label, y_i, is subsequently received, and the model is updated accordingly (x_i, y_i). An URL's features are represented as a vector x and its label as y 1, with +1 indicating a phishing URL and 1 indicating a benign URL. A classification algorithm is given a set of data vectors xi and their labels yi, and it uses this labelled data to train its model. The algorithm is then given a fresh data vector x as input, with the purpose of predicting the label y of this new data using its trained model [37]. We have used the implementation of confidence-weighted learning [38,39] in our experiments to classify our phishing URLs dataset. Also, we have used the implementation of adaptive regularization of weights [40,41] in our experiments to classify our phishing URLs dataset.

4. Experiments and Discussion

4.1. Data source and dataset

From the benchmark sources, we gathered benign and phishing URLs and separated the dataset into a training and testing set with a ratio of 66:34, i.e. 66 percent for training and 34 percent for testing. The Alexa Top sites [42] provided the collection of benign URLs. From the given source of benign URLs, we gathered 1,75,202 benign URLs. We used URLs from two benchmark sources for the phishing dataset: the malware and phishing blacklist of the PhishTank database of validated phishing pages Phishtank [43] and the OpenPhish-Phishing Intelligence OpenPhish [44]. From the above benchmark sources, we gathered 1,75,202 phishing URLs. Table 6 shows the split of the dataset.

Table 6. Training and testing datasets

Class of URLs	Training samples	Testing samples	Total samples
Benign samples	1,15,634	59,568	1,75,202
Phishing samples	1,15,634	59,568	1,75,202
Total samples	2,31,268	1,19,136	3,50,404

4.2. Evaluation Metrics

The confusion matrix provided in Table 7 can be used to assess the correctness of a binary classification problem. To assess the performance of the classifiers, we derived the following measures. A binary classifier assigns a positive or negative label to all data items in a test dataset. True positive (tp), true negative (tn), false positive (fp), and false negative (fn) are the four results of this categorization (or prediction) [45].

Table 7. Confusion Matrix

		Predicted	
		Positive	Negative
Actual	Positive	tp	fn
	Negative	fp	tn

The binary performance evaluation measures like accuracy, FPR, FNR, precision, recall, and f-measure are given as below,

$$Accuracy = \frac{Tp + Tn}{Tp + Tn + Fn + Fp} \quad (4)$$

$$FPR = \frac{Fp}{Tn + Fp} \quad (5)$$

$$FNR = \frac{Fn}{Tp + Fn} \quad (6)$$

$$Precision = \frac{Tp}{Tp + Fp} \quad (7)$$

$$Recall = \frac{Tp}{Tp + Fn} \quad (8)$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (9)$$

4.3. Performance evaluation on our binary URL dataset

We have evaluated online learning classifiers like confidence weighted learning (CW) and Adaptive Regularization of Weight Vectors (AROW) on our three subsets of dataset like dataset using lexical features, using string complexity measures and combined dataset using lexical and string complexity measures. Table 8 and Figure 4 shows the performance analysis of online learning classifiers on our balanced dataset using lexical features. Here, Confidence weighted learning (CW) classifier achieved 97.89% of accuracy on test set, error-

rate of 2.11%, FPR of 0.001, FNR of 0.039, precision of 99.85%, recall of 96.08% and F-measure of 0.979. The Adaptive Regularization of Weight Vectors (AROW) classifier achieved 91.72% of accuracy on test set, error-rate of 8.28%, FPR of 0.085, FNR of 0.081, precision of 91.50%, recall of 91.90% and F-measure of 0.917.

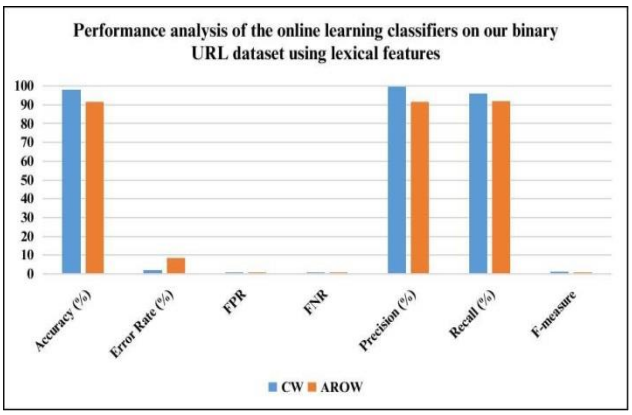

Figure 4. Using lexical features, we evaluated the performance of online learning classifiers on our binary URL dataset

Table 9 and Figure 5 shows the performance analysis of online learning classifiers on our balanced dataset using URL string complexity measure features. Here, CW classifier achieved 89.07% of accuracy on test set, error-rate of 10.93%, FPR of 0.110, FNR of 0.107, precision of 88.89%, recall of 89.21% and F-measure of 0.890. The AROW classifier achieved 90.91% of accuracy on test set, error-rate of 9.09%, FPR of 0.008, FNR of 0.149, precision of 99.0%, recall of 82.60% and F-measure of 0.900.

Table 10 and Figure 6 shows the performance analysis of online learning classifiers on our combined balanced dataset using lexical and string complexity features. Here, Confidence weighted machine learning (CW) classifier achieved 98.35% of accuracy on the test set, error-rate of 1.65%, FPR of 0.026, FNR of 0.005, precision of 97.29%, recall of 99.41% and F-measure of 0.983. The Adaptive Regularization of Weight Vectors(AROW) classifier achieve 97.28% of accuracy on test set, error-rate of 2.72%, FPR of 0.000, FNR of 0.052, precision of 100%, recall of 94.60% and F-measure of 0.972.

Table 8. Lexical Features were used to analyse the performance of online learning classifiers on our binary URL dataset

Classifier	Accuracy (%)	Error Rate (%)	FPR	FNR	Precision (%)	Recall (%)	F-measure
CW	97.89	2.11	0.001	0.039	99.85	96.08	0.979
AROW	91.72	8.28	0.085	0.081	91.50	91.90	0.917

Table 9. Using string complexity features, we analysed the performance of online learning classifiers on our binary URL dataset

Classifier	Accuracy (%)	Error Rate (%)	FPR	FNR	Precision (%)	Recall (%)	F-measure
CW	89.07	10.93	0.110	0.107	88.89	89.21	0.890
AROW	90.91	9.09	0.008	0.149	99.0	82.60	0.900

Table 10. On our binary URL dataset (Lexical + String complexity characteristics), we performed a detailed performance analysis of the online learning classifiers

Classifier	Accuracy (%)	Error Rate (%)	FPR	FNR	Precision (%)	Recall (%)	F-measure
CW	98.35	1.65	0.026	0.005	97.29	99.41	0.983
AROW	97.28	2.72	0.000	0.052	100	94.60	0.972

Table 11. Comparative performance evaluation of our work with CANTINA+

Approaches	#Features	Precision(%)	Recall(%)	F-measure
CANTINA+ [44]	15	97.5	93.47	0.963
Our work using CW	15	97.29	99.41	0.983
Our work using AROW	15	100	94.60	0.972

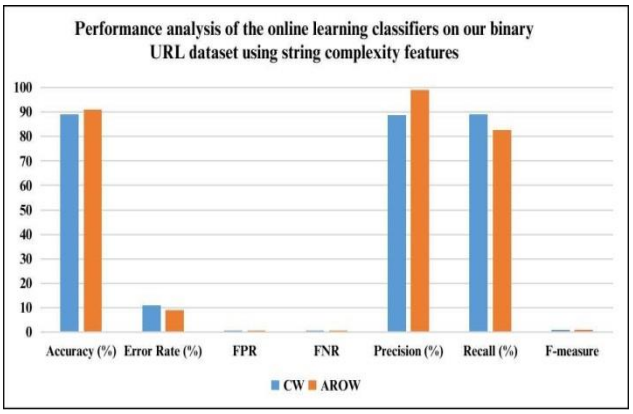

Figure 5. Using string complexity features, we analysed the performance of online learning classifiers on our binary URL dataset

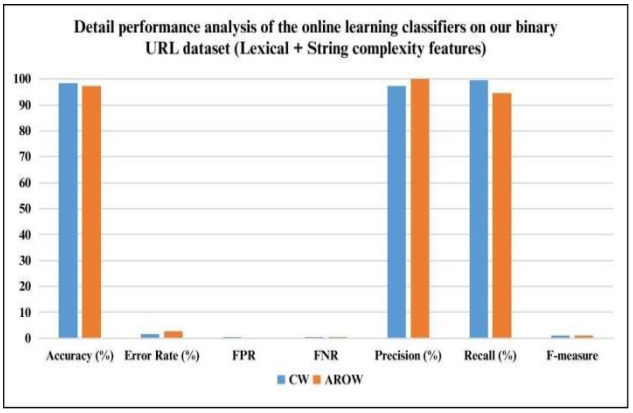

Figure 6. On our binary URL dataset (Lexical + String complexity characteristics), we performed a detailed performance analysis of the online learning classifiers

Table 11 illustrate the comparative performance evaluation of our approach with other phishing URLs detection approach [46]. As shown in Table 11, the metrics of our approach are comparatively better than other approach, except [46] achieved better precision than our approach using CW.

5. Discussion

5.1. Positive impacts of Phishing Detection System

Following are some of the positive impacts of Phishing Detection System.

- Eliminate the cyber threat risk level.
- Increase user alertness to phishing risks.
- Instill a cyber-security culture and create cyber security heroes.
- Change behaviour to eliminate the automatic trust response.

5.2. Negative impacts of Phishing Detection System

Following are some of the negative impacts of Phishing Detection System. Phishing has a list of negative effects on a business including,

- Loss of money,
- Loss of intellectual property,
- Damage to reputation, and
- Disruption of operational activities.

5.3. Impacts of Phishing Detection System on society

- Phishing emails can directly reach millions of people and hide amid the massive quantity of innocent emails that busy individuals receive.
- Malware (such as ransomware) can be installed, systems can be damaged, or intellectual property and money can be stolen.
- Phishing emails may affect any size or kind of organisation.

5.4. Benefits for the academic community and government

Following are the benefits of Phishing Detection System for the academic community and government.

- Academic community can use Phishing Detection System as a research platform for further study to solve open challenges in this field.
- They can provide more significant solutions to detect the new threats in an efficient way with minimum false positives and false negatives.
- Government agencies and organizations can effectively use Phishing Detection System to detect the threats before they can expose the systems.

6. Conclusions

Based on lexical and string complexity analysis, this paper proposed a static method for distinguishing between benign and phishing URLs. Only 15 discriminative URL features, such as lexical features, entropy, Kolmogorov complexity, and Huffman coding complexity, were used. A large binary labelled balanced dataset of phishing and benign URLs with 1,75,202 benign and 1,75,202 phishing URLs is prepared. The results of our experiments on our balanced binary dataset show that our approach is effective for detecting phishing URLs. Two cutting-edge supervised online learning classifiers are used to evaluate the proposed approach. The CW classifier achieved detection accuracy of 98.35%, error rate of 1.65%, FPR of 0.026, FNR of 0.005, precision of 97.29%, recall of 99.41%, and f-measure of 0.983. AROW classifier also achieved 97.28% accuracy, 2.72% error rate, 0.000 FPR, 0.052 FNR, 100% precision, 94.60% recall, and 0.972 f-measure. The results of the experiments suggest that the proposed method enhances phishing URL detection while using the fewest system resources.

References

[1] Anti-Phishing Working Group (APWG) Phishing Activity Trends Report, 1st Quarter 2021, Anti-Phishing Working Group, Inc. (2021), https://docs.apwg.org/reports/apwg_trends_report_q1_2021.pdf.

[2] PhishLabs Threat Trends and Intelligence Report Q1 2021, PhishLabs, https://info.phishlabs.com/q1-2021-threat-trends-intelligence-report

[3] Sahoo, D., Liu, C., Hoi, S. C., and Solouk, V. Malicious URL Detection using Machine Learning: A Survey, arXiv preprint arXiv:1701.07179. 2019, 1–37.

[4] Zabihimayvan, M., Doran, D. and Solouk, V. Fuzzy Rough Set Feature Selection to Enhance Phishing Attack Detection, arXiv preprint: 1903.05675. (2019) 1–6.

[5] Ding, Y., Luktarhan, N., Li, K. and Slamu, W. A keyword-based combination approach for detecting phishing web pages, Computers & Security. 84, 2019, 1–6, doi:10.https://doi.org/10.1016/j.cose.2019.03.018.

[6] Niakanlahiji, A., Chu, B. T. and Al-Shaer, E. PhishMon: A Machine Learning Framework for Detecting Phishing Web pages, In : IEEE Int. Conf. Intelligence and Security Informatics (ISI), (Miami, FL, USA, 2018), pp. 220–225.

[7] Yuan, H., Chen, X., Li, Y., Yang, Z. and Liu, W. Detecting Phishing Websites and Targets Based on URLs and Webpage Links, In: Int. Conf. Pattern Recognition (ICPR), (Beijing, China, 2018), pp. 3669–3674.

[8] Babagoli, M., Aghababa, M.P, M.P. and Solouk, V. Heuristic nonlinear regression strategy for detecting phishing websites, Soft Computing. 23(12), 2019, 4315–4327.

[9] Arab, M., and Sohrabi, M. K. Proposing a new clustering method to detect phishing web- sites, Turkish Journal of Electrical Engineering and Computer Sciences. 25(6), 2017, 4757–4767.

[10] Patil, D. R.Patil and J. B. Feature-based Malicious URL and Attack Type Detection Using Multi-class Classification, The ISC International Journal of Information Security (ISeCure). 10(2), 2018, 141–162, doi:10.22042/ISECURE.2018.0.0.1.

[11] Basnet, R., Mukkamala, S. and Sung, A. H. Detection of phishing attacks: A machine learning approach, Soft Computing Applications in Industry, 2008, 373–383.

[12] Mishra, A. and Gupta, B. B. Intelligent phishing detection system using similarity matching algorithms, International Journal of Information and Communication Technology. 12(1-2), 2018, 51–73.

[13] HR, M. G., Adithya, M. V. and Vinay, S. Development of anti-phishing browser based on random forest and rule of extraction framework, Cybersecurity. 3(1), 2020, 1–14.

[14] Adebowale, M. A., Lwin, K. T., Sanchez, E. and Hossain, M. A. Intelligent web- phishing detection and protection scheme using integrated features of Images, frames and text, Expert Systems with Applications. 115, 2019, 300–313.

[15] Cooper, M., Levy, Y., Wang, L. and Dringus, L. Heads-up! An alert and warning system for phishing emails, Organizational Cybersecurity Journal: Practice, Process and People, 2021, 1–22.

[16] Evans, K., Abuadbba, A., Ahmed, M., Wu, T., Johnstone, M., and Nepal, S. RAIDER: Reinforcement-aided Spear Phishing Detector, arXiv preprint arXiv:2105.07582, 2021, 1–19.

[17] Mohammada, G. B., Shitharthb, S. and Kumarc, P. R. Integrated Machine Learning Model for an URL Phishing Detection, International Journal of Grid and Distributed Computing, 14(1) , 2020, 513–529.

[18] Jain, A. K. and Gupta, B. B. Phishing detection: analysis of visual similarity based approaches, Security and Communication Networks, 2017, 1–20.

[19] Mourtaji, Y., Bouhorma, M., Alghazzawi, D., Aldabbagh, G.andAlghamdi, A. Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network, Wireless Communications and Mobile Computing, 2021, 1–24.

[20] Akdemir, N. and Yenal, S. How Phishers Exploit the Coronavirus Pandemic: A Content Analysis of COVID-19 Themed Phishing Emails, SAGE Open, 11(3), 2021, 1–14,doi: 21582440211031879.

[21] El Aassal, A., Baki, S., Das, A. and Verma, R. M. An in-depth benchmarking and evaluation of phishing detection research for security needs, IEEE Access, 8, 2020, 22170–22192, doi: 10.1109/ACCESS.2020.2969780.

[22] Sahingoz, O. K., Buber, E., Demir, O. and Diri, B. Machine learning based phishing detection from URLs, Expert Systems with Applications, 117, 2019, 345–357, doi: https://doi.org/10.1016/j.eswa.2018.09.029.

[23] Butnaru, A., Mylonas, A. and Pitropakis, N. Towards Lightweight URL-Based Phishing Detection,

Future Internet. 13(6), 2021, 1–15, doi: https://doi.org/10.3390/fi13060154.
[24] Bagui, S., Nandi, D. and White, R. J. Machine Learning and Deep Learning for Phishing Email Classification using One-Hot Encoding, Journal of Computer Science. 17(7), 2021, 610–623, doi: https://doi.org/10.3844/jcssp.2021.610.623.
[25] Zhu, E., Chen, Y., Ye, C., Li, X. and Liu, F. OFS-NN: an effective phishing websites detection model based on optimal feature selection and neural network, IEEE Access. 7, 2019, 73271–73284, doi: 10.1109/ACCESS.2019.2920655.
[26] Ali, W. and Malebary, S. Particle swarm optimization-based feature weighting for improving intelligent phishing website detection, IEEE Access, 8, 2020, 116766–116780, doi: 10.1109/ACCESS.2020.3003569.
[27] Mao, J., Bian, J., Tian, W., Zhu, S., Wei, T., Li, A., and Liang, Z. Phishing page detection via learning classifiers from page layout feature, EURASIP Journal on Wireless Communications and Networking, 1, 2019, 1–14, doi: https://doi.org/10.1186/s13638- 019-1361-0.
[28] Acharya, B. and Vadrevu, P. PhishPrint: Evading Phishing Detection Crawlers by Prior Profiling, In: 30th USENIX Security Symposium, 2021, 3775–3792.
[29] Rasool, R. U., Ahmed, K., Anwar, Z., Wang, H., Ashraf, U., & Rafique, W. CyberPulse++: A machine learning-based security framework for detecting link flooding attacks in software defined networks. International Journal of Intelligent Systems, 36(8), 2021, 3852-3879.
[30] Vimalachandran, P., Liu, H., Lin, Y., Ji, K., Wang, H., & Zhang, Y. Improving accessibility of the Australian My Health Records while preserving privacy and security of the system. Health Information Science and Systems, 8(1), 2020, 1-9.
[31] Patil, D. R., Patil, J. B. Malicious web pages detection using feature selection techniques and machine learning, International Journal of High Performance Computing and Networking., 14(4), 2019, 473–488., doi: 10.1504/IJHPCN.2019.102355.
[32] Patil, D. R., Patil and J. B. Malicious URLs detection using decision tree classifiers and majority voting technique, Cybernetics and Information Technologies, 18(1), 2018, 11–29, doi: https://doi.org/10.2478/cait-2018-0002.
[33] Verma R., Das A, What's in a URL: Fast Feature Extraction and Malicious URL Detection, In: 3rd International Workshop on Security and Privacy Analytics, (Scottsdale, AZ, United States, 2017, 55–63.
[34] Evans, S. C., Hershey, J. E and Saulnier, G. Kolmogorov complexity estimation and analysis, In: Sixth World Conference on Systemics, Cybernetics and Informatics, (Orlando, Fla,), 2002.
[35] Pao, H. K., Chou, Y. L. and Lee, Y. J. Malicious URL detection based on Kolmogorov complexity estimation, In: IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology, (Macau, China), 2012, 380– 387.
[36] Moffat, A. Huffman coding, In: ACM Computing Surveys (CSUR), 2019, 31–35.
[37] Dredze, M., Crammer, K. and Pereira, F. Confidence-weighted linear classification, In: 25th International Conference on Machine learning, (Helsinki Finland), 2008, 264–271.
[38] Dahlmeier D., Ng H. T. and Ng E. J. F. NUS at the HOO 2012 Shared Task, In: Seventh Workshop on Building Educational Applications Using NLP, 2008, 216– 224.
[39] Confidence-weighted (CW) learning. (2019), http://www.comp.nus.edu.sg/nlp/ software.html
[40] Crammer, K., Kulesza, A. and Dredze, M. Adaptive regularization of weight vectors, In: Advances in Neural Information Processing Systems, 2009, 414–422.
[41] AROW++: An implementation of the efficient confidence-weighted classifier. (2019), https://github.com/tetsuok/arowpp
[42] Alexa: Alexa top global websites. (2021), http://www.alexa.com/topsites
[43] Phishtank: Join the fight against phishing. (2021), https://www.phishtank.com
[44] OpenPhish - Phishing Intelligence. (2021), https://openphish.com
[45] Sokolova M. and Lapalme G. A systematic analysis of performance measures for classification tasks, Information Processing and Management, 45(4), 2009, 427–437, doi: 10.1016/j.ipm.2009.03.002.
[46] Xiang, J. Hong, C. P. Rose and L. Cranor. Cantina+: a feature-rich machine learning framework for detecting phishing web sites, ACM Transactions on Information and System Security (TISSEC), 14(2), 2011, 1–28, doi: https://doi.org/10.1145/2019599.2019606.