Journal Homepage: - www.journalijar.com
INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)
Article DOI: 10.21474/IJAR01
DOI URL: http://dx.doi.org/10.21474/IJAR01

*RESEARCH ARTICLE*

**National Conference on Innovation in Science, Engineering and Management (NCISEM-2022)**
**Date of Conference 11-12 March 2022**

# PROPOSED SYSTEM FOR CONTENT BASED MALICIOUS URL DETECTION

**Nitin Bhawarkar[1], Sanjog Ghonge[1], Vaishnavi Dhange[1] and Dr. A. N. Thakre[2]**

1. Research Scholar, Department of Computer Engineering, Bapurao Deshmukh College of Engineering Sewagram, Wardha.
2. Head of Department of Computer Engineering, Bapurao Deshmukh College of Engineering Sewagram, Wardha.

## *Manuscript Info*

***Key words:-***
Malicious URL Detection, Machine Learning

## *Abstract*

Malicious links are used as a source by the distribution channels to spread malware across the Internet, cybercriminals utilize fraudulent URLs as distribution channels. In order to gain remote access to the victim's computer, attackers use browser vulnerabilities to install malware. Most malware is designed to acquire network access, ex-filtrate sensitive data, and secretly monitor computer systems. This research presents a classification based on association (CBA) data mining approach for detecting malicious URLs based on URL and webpage content features.



## Introduction:-

The growth of the World Wide Web (WWW) has attracted hackers' interest in using the web to distribute malware and breach people's and organizations' networks. Attackers utilize scripts, exploits, and executable files to steal credit card information from online stores.

Malicious universal resource locators were recognized as web risks, according to the Kaspersky security report for 2019. (URLs). Malicious websites try to infect victims' computers with malware, acquire sensitive information, and take complete control over their devices. The most common attacks that activate malicious URLs to propagate malware are drive-by downloads and social engineering. An attacker uses malicious client-side scripting code to target a vulnerability in a web browser or plugin in a drive-by download attack. Cybercriminals have revised sophisticated methods of luring people into clicking on infected links and opening suspicious attachments, such as advertising and breaking news. When a user visits the hacked site, the malicious script runs, exploiting a flaw in the web browser to download the malicious Payload, which grants attackers remote access to the victim's machine. In recent years, JavaScript-based attacks have been reported to account for a significant portion of web attacks.

For fighting against malicious web pages, researchers have offered defense mechanisms such as static analysis, dynamic analysis, blacklisting-based, and heuristic-based approaches. Static analysis approaches inspect websites without requiring the page to be rendered in a browser. Cuckoo and SpyProxy are dynamic analytic techniques that leverage a behavior analysis environment to detect malicious scripts. Attackers can quickly recognize the analysis

**Corresponding Author:- Nitin Bhawarkar**
Address:- Research Scholar, Department of Computer Engineering, Bapurao Deshmukh College of Engineering Sewagram, Wardha.

setting, increasing their chances of avoiding the behavioral monitoring procedure. Requested URLs are tested against predetermined dangerous URLs in blacklist-based approaches, but they are not proactive in detecting newly developing malicious web pages. To scan websites, heuristic-based approaches establish signatures of known attack payloads. Unfortunately, attackers can readily avoid systems based on predetermined signatures, and often fail to detect new attacks.

**What is URL?**

A Uniform Resource Locator (URL), colloquially termed a web address, is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. A URL is a specific type of Uniform Resource Identifier, although many people use the two terms interchangeably.

URL is the abbreviation of Uniform Resource Locator, which is the global address of documents and other resources on the World Wide Web. A URL has two main components:

1. Protocol identifier (indicates what protocol to use)
2. Resource name (specifies the IP address or the domain name where the resource is located). The protocol identifier and the resource name are separated by a colon and two forward slashes, e.g. Figure 1.1.

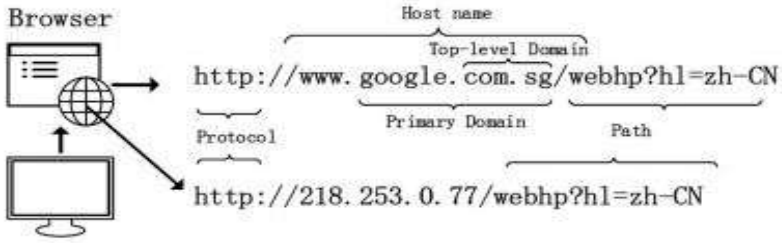


**Fig. 1.1:-** Format of a URL - "Uniform Resource Locator"

Compromised URLs that are used for cyber-attacks are termed as malicious URLs. In fact, it was noted that close to one-third of all websites are potentially malicious in nature, demonstrating rampant use of malicious URLs to perpetrate cyber-crimes.

**So, what is malicious URL?**

**Malicious URL** is a **URL** created with **malicious** purposes, among them, to download any type of malware to the affected computer, which can be contained in spam or phishing messages, or even improve its position in search engines using Blackhat SEO techniques.

Within the multitude of cyber threats out there, malicious websites play a critical role in today's attacks and scams. Malicious URLs can be delivered to users via email, text message, pop-ups or shady advertisements. The end result can often be downloaded malware, spyware, ransom ware, compromised accounts, and all the headaches those threats entail. It should be evident that being aware of what a Malicious URL is, and how it can do damage, is key to your email security.

Simply put, receiving a URL can be similar to a stranger inviting you to their house. Their invitation might promise food and drink, and you could go over for a visit, but you have no idea what will really happen until you walk inside. Who knows – in the best case scenario, there might be homemade lasagna on the table and great company? A more "malicious invite" might mean your wallet will be stolen. It could also lead to you being kidnapped and held for ransom.

This doesn't mean all strangers are out to get you, but when you click on a link from somewhere unexpected, how are you supposed to know where it will take you? You might actually win a prize (highly unlikely), but it is also highly probable it is a malicious URL and you'll end up downloading a virus, malware, get phished or suffer any other scam.

It is also worth noting that sometimes malicious URLs may appear to be coming from a friend, but in many cases this is either them not being aware of what they're sending you, or their email account has been compromised.

## Literature Survey:-

[1] Proposed with an objective to enhance PDF maldoc detection and pro-long the lives of existing analysis and detection tools, we developed an approach to identify a set of features extracted using available tools and derive a new set of features. Our findings were validated with large datasets from VirusTotal. Our analysis shows that by applying our technique, we can reduce the feature-set size by more than 60% while increasing the classification accuracy by 2%.

[2] Proposed model which has been trained using a stored dataset containing close to 1750 URLs and applying different machine learning algorithms ranging from Logistics Regression to Support Vector Machines. After iterations of training and testing, it is found that Random Forest produces the highest accuracy.

[3] Proposed ML-based approaches for detection of phishing websites is conducted. This work presents a comprehensive review of conventional ML techniques which are significant for detection of malicious attacks on websites. Other protection strategies presented in this paper are deep learning, heuristic, and automated techniques.

[4] Identifies a generic features set f or malicious URL classification. We finalized 47 most common significant features out of 106 that have potential to identify any malicious URLs accurately and precisely with the weighted average low false-positive rate of 2.88% and the accuracy of 96.6% for two different datasets.

[5] Approach for capturing semantic information directly from the URL via distributed word representations, and complement that information with general domain-specific features to boost malicious URL detection performance.

[6] The new method for malicious URL detection with fewer number of features extracted only from URL. That reduces execution time and storage requirements. It also highlights the recent research work in the domain and issues in the existing work. Result shows that random forest classifier is outperformed than the other classifiers.

[7] Proposed the previous research and this study is that based on the characteristics of malicious web host information and the characteristics of URL information, this paper uses the word vector model word2vec to train the URL word vector feature, and extracts the "texture fingerprint" feature of the malicious webpages.

**Existing System**

**Methods to detect Malicious URLs**

In this section, I present the key principles used by researchers to solve the problem of malicious URL detection. A variety of approaches have been attempted to tackle the problem of malicious URL detection. These approaches can be broadly grouped into two categories: (i) Blacklisting or Heuristics (ii) Machine Learning.

**Blacklisting or Heuristics method**

The most common method to detect malicious URLs deployed by many antivirus groups is the blacklist method. Blacklists are essentially a database of URLs that have been confirmed to be malicious in the past. This database is compiled over time (often through crowd-sourcing solutions, e.g. Phish Tank), as and when it becomes known that a URL is malicious. Such a technique is extremely fast due to a simple query overhead, and hence is very easy to implement.

Additionally, such a technique would (intuitively) have a very low false-positive rate (although, it was reported that often blacklisting suffered from non-trivial false-positive rates).

Blacklisting is a common and classical technique for detecting malicious URLs, which often maintains a list of URLs that are known to be malicious. Whenever a new URL is visited, a database lookup is performed. If the URL is present in the blacklist, it is considered to be malicious and then a warning will be generated; else it is assumed to be benign. Blacklisting suffers from the inability to maintain an exhaustive list of all possible malicious URLs, as new URLs can be easily generated daily, thus making it impossible for them to detect new threats. Despite several problems faced by blacklisting due to their simplicity and efficiency, they continue to be one of the most commonly used techniques by many anti-virus systems today. Common attacks are identified, and based on their behavior; a signature is assigned to this attack type. However, such methods can be designed for only a limited number of common threats. A more specific version of heuristic approaches is through analysis of execution dynamics of the webpage .Here too, the idea is to look for a signature of malicious activity such as unusual process creation, repeated redirection, etc. These methods necessarily require visiting the webpage and thus the URLs actually can make an attack.

**Machine learning**

Machine Learning approaches, use a set of URLs as training data, and based on the statistical properties, learn a prediction function to classify a URL as malicious or benign. This gives them the ability to generalize to new URLs

unlike blacklisting methods. The primary requirement for training a machine learning model is the presence of training data. In the context of malicious URL detection, this would correspond to a set of large number of URLs. Machine learning can broadly be classified into supervised, unsupervised, and semi-supervised, which correspond to having the labels for the training data, not having the labels, and having labels for limited fraction of training data, respectively. Labels correspond to the knowledge that a URL is malicious or benign.

These approaches try to analyze the information of a URL and its corresponding websites or web pages, by extracting good feature representations of URLs, and training a prediction model on training data of both malicious and benign URLs. There are two-types of features that can be used - static features, and dynamic features. In static analysis, we perform the analysis of a webpage based on information available without executing the URL .The features extracted include lexical features from the URL string, information about the host, and sometimes even HTML and JavaScript content. Since no execution is required, these methods are safer than the dynamic approaches. The underlying assumption is that the distribution of these features is different for malicious and benign URLs. Using this distribution information, a prediction model can be built, which can make predictions on new URLs. Dynamic analysis techniques include monitoring the behavior of the systems which are potential victims, to look for any anomaly. These include which monitor the system call sequences for abnormal behavior, and which mine internet access log data for suspicious activity. Dynamic analysis techniques have inherent risks, and are difficult to implement and generalize. In this report, we shall focus on static techniques and mainly the simplest, logistic regression.

**Proposed System**

**Malicious URL Detection Tools**

a) Data Extraction: Data extraction is the act or process of retrieving data out of data sources for further data processing or data storage (data migration). The import into the intermediate extracting system is thus usually followed by data transformation and possibly the addition of metadata prior to export to another stage in the data workflow.
b) Data cleaning: Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset. But it is crucial to establish a template for your data cleaning process so you know you are doing it the right way every time.
c) Stop Word Removal: Stop word removal is the process of converting data to something a computer can understand is referred to as pre-processing. One of the major forms of pre-processing is to filter out useless data. In natural language processing, useless words (data), are referred to as stop words. A stop word is a commonly used word (such as “the”, “a”, “an”, “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.
d) Negative keywords datasets: Negative keywords can be taken from Wordstream. We can use other dataset as well.

The Naive Bayes Classifier is a kind of probabilistic classification mechanism rooted in the Bayesian Theorem which is a posthumous theory of Thomas Bayes. From the perspective of classification, the main goal is to find the best mapping between a piece of new data and a set of classifications within a particular problem domain. For the purpose of making this mapping probabilistically computable, some mathematical manipulations are performed to transform joint probabilities into the multiplications of prior probabilities and conditional probabilities. As a machine learning and data mining approach, this mathematical transformation is kind of unnatural and might be hard for beginners to comprehend because it is turning a simple division into a long series of numerators divided by another long series of denominators. However, these unnatural transformations are necessary as prior probabilities and conditional probabilities are easy to summarize from a given data set by simply counting the number of instances with or without a given condition.

**The Proposed Algorithm based on Naive Bayes**

**INPUT:** Training set, URLs to be tested.
**OUTPUT:** Testing content present on webpage.

**Step 1:** For given feature calculate its sub features for training purpose using the training set.

**Step 2:** The classifier is created from the training set using a Gaussian distribution and by calculating mean and variance of each sub feature.

**Step 3:** Probability of individual class is calculated.

**Step 4:** Testing sample with their calculated feature is taken for classification.

**Step 5:** Posterior for each class is calculated.

**Step 6:** Analyse posterior values of each class.

**Step 7:** Among Four classes, class with greater value of posterior is assigned to testing domain.

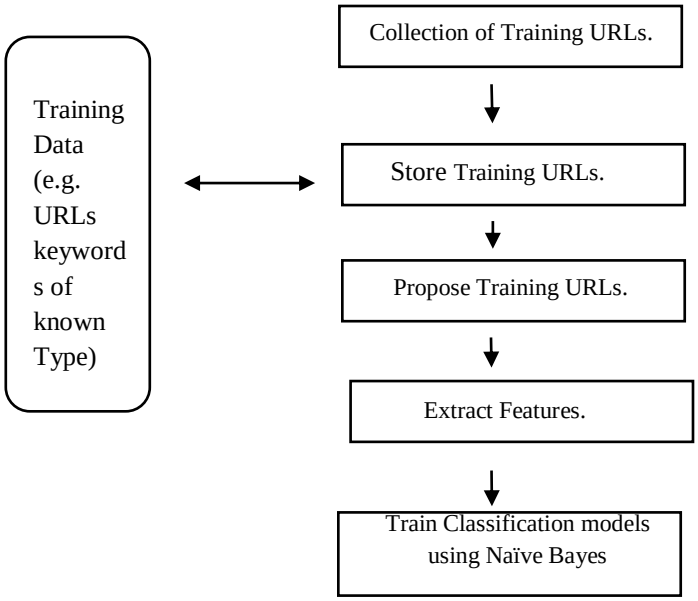


**Fig 1:-** Training of Data.

**Architectural Diagram**

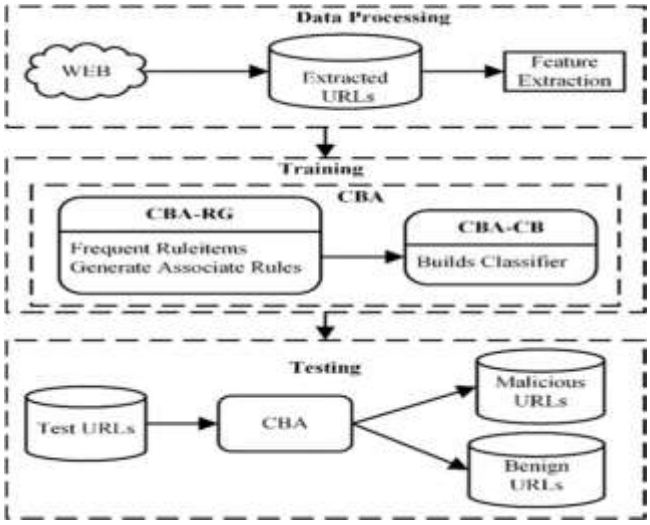


**Fig 2:-** System Architecture of malicious URL detection.

**Flow Chart**

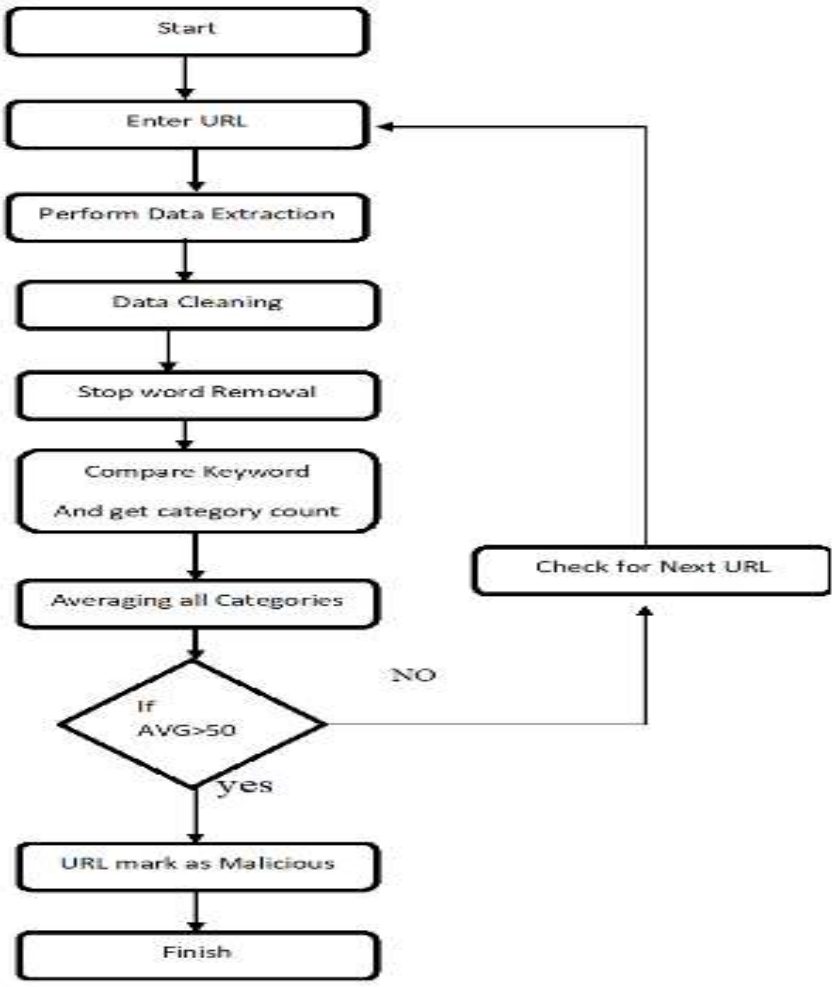


**Fig 3:- Flowchart of System**

## Objectives Of System:-

The inclusion of training data is the most important condition for training a machine learning model. This would correspond to a huge number of URLs in the context of malicious URL detection. Machine learning is divided into three categories: supervised, unsupervised, and semi-supervised, which correspond to having labels for the training data, not having labels, and having labels for a small portion of the training data. The information about whether a URL is harmful or benign is represented by labels.

By extracting good feature representations of URLs and building a prediction model using training data of both harmful and benign URLs, these approaches attempt to assess the information of a URL and its accompanying websites or web pages. Static features and dynamic features are the two sorts of features that can be used. Static analysis is when we analyse a webpage using information we already have rather than executing the URL.

Lexical aspects from the URL string, information about the host, and sometimes even HTML content are among the features extracted. These methods are safer than dynamic ones since they do not require execution. The basic premise is that for dangerous and benign URLs, the distribution of these properties differs. A prediction model that can make predictions on new URLs can be developed using this distribution information. One of the ways utilised in dynamic analysis is to monitor the activity of possible victims' systems in order to look for irregularities. These are programmes that scan system call sequences for anomalous behaviour and mine internet access log data for

suspicious activity. Dynamic analytic methods are difficult to implement and generalise, and they are fraught with dangers.

## Methodology:-

The dataset to use is obtained a list of URLs with good/bad labels that indicate if the URL in concerned is malicious or not. We will use the good URLs to train the auto-encoder. And it's exciting to see that the model works, so use both good and bad URLs for testing. 80% of the data that will be used for training the model after keeping in the normal cases and 20% will be used for testing purposes.

To train a classifier using only one class (normal cases) is to use an auto-encoder neural network. Auto-encoders are a variety of deep neural networks that aims to produce in their outputs the same data they receive as input. The concept is to feed a non-malicious URL into an auto-encoder, and since we should obtain the identical data in the output, we'll calculate the reconstruction error (the difference between the input and the prediction) to determine whether it's about a malicious URL based on a predefined threshold.
We'll try to break down the entire model into three distinct modules:

The important modules include,
1. Web Scrapping
2. Content Sorting
3. Stop Word Removal
4. Keyword Generation
5. Implement Naive Bayes Algorithm to solve problem present in text classification.
6. Judgement on basis of result.

## Conclusion And Future Scope:-

In this article, we proposed Content Based Malicious URL Detection in which first we take the URL. The data will be extracted by web crawling method. Extracted content keywords compared with Pre-Existing Dataset Keywords. On the judgement of Pre-Existing Negative Keywords compared with Content Keyword. The website is may be Malicious or May not be Malicious.

In the future, we can extend this work by developing a web browser extension for real-time detection of URLs as malicious or benign. Such an extension will notify that in the URL how many Negative Keywords are present. On our requirements, we modified and change the conditions. Basis on that we conclude the website are malicious or not.

## References:-

1. Zhu E, J. Y. (2020). An artificial neural network phishing detection model based on decision tree and optimal features. *Applied Soft Computing*.
2. A, S. (2020). Fundamentals of recurrent neural network and long short-term memory. *Phys D Nonlinear Phenom*.
3. Bhagwat, S. D. (2019). An Implemention of a Mechanism for Malicious URLs Detection. *2019 6th International Conference on Computing for Sustainable Global Development*, pp. 1008–1013.
4. Ahmed Falah, L. P. (2020). Improving Malicious PDF classifier with feature engineering: A data-driven approach. *elsevier*, 13.
5. Akiyama M, Y. T. (2017). Analyzing the ecosystem of malicious URL redirection through longitudinal observation from honeypots. *Comput Sec*.
6. Akshay Sushena Manjeri, K. R. (2019). A Machine Learning Approach for Detecting Malicious Websites using URL Features. *ieee*, 7.
7. W. H. (2019). An algorithm based on bi-IndRNN for analysis and detection of malicious URL. *J. Xinjiang Univ., Natural Sci. Ed*, vol. 36, no. 154, pp. 174-181.
8. Y. P. (2019). Detecting web attacks with end-to-end deep learning. *Journal of Internet Services and Applications*, vol. 10, no. 1, pp. 1-22.
9. A. Z. (2020.). Phishing web site detection using diverse machine learning algorithms. *The Electronic Library*.
10. Aleroud A, Z. L. (2017). 2017) Phishing environments, techniques, and countermeasures: a survey. *Comput Secure 68*, pp. 160-196.

11. Ammar Odeh, I. K. (2021). Machine LearningTechniquesfor Detection of Website Phishing: A Review forPromises and Challenges. *ieee*, 6.
12. Bahnsen AC, B. E. (2017). Classifying phishing urls using recurrent neural networks. *APWG symposium on electronic crime research*, pp 1–8.
13. Basit A, Z. M. (2020). A comprehensive survey of ai-enabled phishing attacks detection techniques. *Telecommun Syst,*, pp 1–16.
14. Benavides E, F. W. (2020). Classification of phishing attack solutions by employing deep learning techniques: A systematic literature review. *Springer*, pp 51–64.
15. BoreGowda, G. H. (2020). Phishing website detection based on effective machine learning approach. *Journal of Cyber Security Technology*, pp. 1-14.
16. Chen, Y. C. (2019). AI@ntiPhish- Machine Learning Mechanisms for Cyber-Phishing Attack. *IEICE Transactions on Information and Systems*, vol. E102-D, no.5, pp. 878–887.
17. Dargan S, K. M. (2020). A survey of deep learning and its applications: a new paradigm to machine learning. *Arch Comput Methods Eng 27*, pp. 1071–1092.
18. F. Feng, Q. Z. (2018.). The application of a novel neural network in the detection of phishing websites. *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-15.
19. Hafiz Mohammd Junaid Khan, Q. N. (2019). Identifying Generic Features for Malicious URL. *ieee*, 6.
20. Huan-huan Wang, L. Y.-w.-f.-j. (2019). Bidirectional LSTM Malicious webpages detection algorithm based. *springer*, 11.
21. Immadisetti Naga Venkata Durga Naveen, M. K. (2019). Detection of Malicious URLs using Machine Learning Techniques. *IJITEE*, 5.
22. K. L. Chiew, C. L. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences*, vol. 484, pp. 153-166,.
23. Karunakaran, P. (2020). Deep Learning Approach to DGA Classification for Effect ive Cyber Security. *Journal of Ubiquit ous Comput ing and Communicat ion Technologies*, 203-213.
24. Le H, P. Q. (2018). URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection.
25. Luo S, S. T. (2018). Android malicious code classification using deep belief network. *KSII Trans Intern Inform System*, pp. 454-475.
26. N. Azeez, B. S. (2020). Identifying phishing attacks in communication networks using URL consistency features. *Int. J. Electron. Secur. Digit. Forensics*, vol. 12, no. 2, p. 200.
27. Nunes, P. .. (2018.). Benchmarking St atic St udy T ools for Web Security Transact ions on Reliability. Vol. 67, No.3, pp.1159-1175.
28. O. K. Sahingoz, E. B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, vol. 117, pp. 345-357.
29. OlalereM, A. M. (2017). Identification and evaluation of discriminative lexical features of malware URL for real-time classification. *ieee*, pp. 90–95.
30. P eng Yang, e. a. (2019.). P hishing Websit e Exposure based on Mult idimensional Features driven by Deep Learning. *IEEE*.
31. Role of Machine Learning Algorithms Intrusion Detect ion in WSNs: A Survey. (2020). *Journal of Informat ion Technology 2*, pp. 161-173.
32. Sahoo D, L. C. (2017). Malicious URL Detection using Machine Learning: A Survey.
33. Saleem Raja A, V. R. (2021). Lexical features based malicious URL detection using machine learning. *elsevier*, 4.
34. Sara Afzal, M. A. (2021). URLdeepDetect: A Deep Learning Approach for Detecting Malicious URLs Using Semantic Vector Models. *springer*, 27.
35. Shakya, S. a. (2020). Intelligent and adaptive multi-objective optimization in WANET using bio inspired algorithms. *J Soft Comput Paradigm (JSCP)*, pp. 13-23.
36. Shengwei T, X. Z. (2018). Causal relationship extraction based on bidirectional LSTM in Uighur language Electron Inf Technology. pp. 200–208.
37. Soleimani, F. A. (2019). An effective feature selection method for web spam detection. *Knowledge-Based Systems*, vol. 166, pp. 198 – 206.
38. T. Li, G. K. (2020). Improving malicious urls detection via feature engineering: Linear and nonlinear space transformation methods. *Information Systems*, vol. 91, p. 101494.
39. Tejpal Sharma, D. R. (2021). Malicious application detection in android- A. *elsevier*, 33.
40. Tejpal Sharma, D. R. (2021). Malicious application detection in android—A systematic literature. *elsevier*, 33.

41. W. Wei, Q. K. (2020). Accurate and fast URL phishing detector:Aconvolutional neural network approach. *Computer Network*, vol. 178.
42. X. D. Hoang. (2018). A website defacement detection method based on machine learning techniques. *Proceedings of the Ninth International Symposium on Information and Communication Technology*, pp. 443-448.
43. Y. F. Peng, S. W. (2019). A Joint Approach Approachto Detect Malicious URL Based on Attention Mechanism. *International Journal of Computational Intelligence and Applications*, pp. 18.
44. Y. Huang, J. Q. (pp. 22-26.). Phishing URL detection via capsule-based neural network. *IEEE 13th Int. Conf. Anti-Counterfeiting, Secure Identification* , 2019.
45. Y. Huang, Q. Y. (2019). `Phishing URL detection via CNN and attention-based hierarchical RNN. *18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, pp. 112-119.
46. Y. Li, Z. Y. (2019.). A stacking model using URL and HTML features for phishing webpage detection. *Future Gener.Comput. System*, vol. 94, pp. 27-39,.
47. Y. Liang, J. D. (2019). Bidirectional LSTM: An innovative approach for phishing URL identification. *Proc. Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput.*, pp. 326-337.
48. Yiran Ma, Q. G. (2021). Malicious URL Classification Model Based on Improved Sparrow Search Algorithm. *ieee*, 5.
49. Yongjie Huang, J. Q. (2019). Phishing URL Detection Via Capsule-Based Neural Network. *ieee*, 5.
50. Zhuo, Z. e. (2018). Websit e Fingerprinting At tack on Anonymity Networks Based on t he P rofile Hidden Markov Ideal. *Information Forensics and Security*, Vol. 13, No. 5, pp.1081-1095.