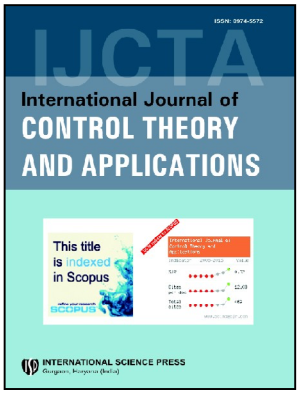

International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 10 • Number 29 • 2017

A Feature Based Framework to Detect Malicious URLS

N. Jayakanthan[1*] A.V. Ramani[2] and M. Ravichandran[2]

[1]*Department of Computer Applications, Kumaraguru College of Technology, Coimbatore, Tamilnadu, India.*
[2]*Department of Computer Science, Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore, Tamilnadu, India.*
[1*]*Corresponding Author Email: haijai2@gmail.com*

Abstract: The main security issue in web application is malicious URL. It redirects the user to an unsafe website which contains malicious content. Malicious URLs exploits the vulnerabilities present in browser to attack the system that leads to loss of information. This is a major threat with current internet having billions of web sites. In this paper, a novel approach for detecting and preventing malicious URLs is proposed. The proposed system is capable of detecting the URL attacks which contains the suspicious special characters, domains, sub domains and path. The enhanced ID3 decision tree algorithm is used for detection. This light weight approach detect the malicious website with low performance overhead of client machine. The algorithm is validated with real time data. The results are found to be encouraging when compared with lexical feature based phishing URL detection, efficient malicious URL based on feature classification and malicious URL detection algorithm based on BM pattern matching methods.

Keywords: URL, phishing, malicious domain, web security, machine learning.

1. INTRODUCTION

The cyber attacks are highly dynamic. Criminals use the web as a platform for a wide range of attacks. The attacker creates a fake web page which looks like the website of trustworthy organization and enable the user to perform his regular operation like e-banking etc. By using these fake websites, the attacker access the secured information like usernames and passwords of credit/debit cards, banking etc., The Water marking techniques used to prove the copy right of the software[4] are used by criminals to inject the malicious code in the software. The RSA Anti-Fraud Command [21] has reported more than 275,000 phishing attacks and is a key industry source for intelligence on new and emerging online threats. Varieties of techniques are available to detect malicious URLs like lexical feature based phishing URL detection, web wallet, honey pots and web crawlers. But many malicious websites are still not properly detected and also report a high percentage of false negative (FN) and false positive (FP).Several algorithms have been proposed in literature to detect malicious websites. These algorithms belong to various categories like detection of phishing attacks [1], email filtering [2], malicious web page identification [3] and malicious behaviour analysis.

In this paper we proposed dynamic anomaly detector for web application based on ID3 decision tree algorithm. It detects the malicious URLs and warns the user about the nature of the attack. Our approach checks the unusual special characters (!’,’@’,’#’,’$’,’^’,’*’,’(‘,’)’,’+’,’{‘,’}’) which are generally the part of the URL that leads to malicious attack. It parses the URLs into domain, sub domain and path and analyse various features of each component to detect the malicious attack. If suspicious features are found, the system blocks the URL. In addition to that our approach checks the URLs membership in the blacklisted profile. The proposed approach accurately blocks the undesirable web pages and the blocking mechanism is performed in the client machine by discriminating the URLs. The proposed approach is a light weight approach for blocking malicious web pages than those methods which employs the content analysis .The drawbacks of such approaches are

1) The entire web pages should be downloaded and analysed.
2) It is a time consuming process which affects the performance of the client machine.

The system is trained using machine learning method. It detects all popular URL attacks. Large set of features is added to detect fake URLs which resembles the original website with slight variations (e.g.:www.goole.com instead of www.google.com) and redirects the user to a phishing website. The patterns and filenames are manipulated to create web attacks and access the secured data. (e.g.darkartsmedia.com/Google.html).

The contributions of the proposed approach are as follows:

1) The proposed approach is a lightweight approach that detects the malicious URLs with low performance over head.
2) This approach introduces novel features based on special characters, domain, sub domain and path to identify the genuine and malicious web pages.
3) This approach is designed, implemented and evaluated over a large data set of malicious and genuine web pages and demonstrated the effectiveness.

The proposed system is evaluated across 600 URLs by analysing the suspicious special characters, domains, sub domains and paths.

The paper is organized as follows. Section 2 reviews the related work. Section 3 presents the architecture of the system. Section 4 reports the methodology. The proposed algorithm is explained in section 5. In section 6 experiment setup and results are reported. Section 7 draws the conclusion.

2. RELATED WORK

Aaron Blum et al [1] developed a lexical based phishing URL detection method to detect phishing domains. These systems detect threats and protect the web from zero hour threat. But the proposed system uses various features to detect a variety of malicious URLs which are not detected by Aaron’s approach. Anderson,1992[2] introduced the concept of computer intrusion detection system. He strongly asserts the needs of such systems. Fillipo Ricca and Paolo Tonella [7] developed an automated tool to detect the anomalies associated with the web page structure and multilingual problems. This approach is not capable to detect the anomalies associated with URLs. Guan et al [9] developed a method to detect malicious URLs in instant messaging. The parameters are anomalies of URL message and behaviour of the sender. A scoring model is used to evaluate the significance of anomaly. Guang Xian and Jason I Hong [10] used information extraction (IE) and Information retrieval (IR) techniques in their hybrid phishing detection system. The system is capable of detecting phishing web pages by differentiating the web pages actual identity and the identity they are imitating. Both the systems are not efficient in detecting the URLs containing malicious sub domains. But the proposed system addresses this issue.

Various algorithms like similarity measures used for pattern recognition [5] are also used to detect the malicious websites. Hyunsang Chou et al [13] proposed an approach to detect malicious web links using machine learning method which uses the textual properties and link structures as the parameters for analysis.

This system uses limited features and is not able to detect malicious URLs web page. Jo˜ao Paulo Magalh˜aes and Luis Moura Silva[14] developed an approach which targeted for web-based and component based applications. It makes use of Aspect-Oriented Programming (AOP) based monitoring, data correlation techniques and time-series alignment algorithms to spot the occurrence of performance anomalies avoiding false alarms due to workload variations. The proposed system reports low false positive and false negative rate than this method. Seifert et al, 2008[23] presents a novel classification method that identifies malicious web pages based on the server relationships involved in rendering a web page. This classification method does not require the dedicated environment that a client honey pot does. In addition, the method is generic in its ability to identify malicious web pages. Any client capable of rendering a web page can be used to interact with the malicious web server. But this approach is not effective in detecting the pages directed by malicious URLs.

Ying Pan and Xuhua Ding [24] examined the anomalies in web pages, in particular, the discrepancy between a web site's identity and its structural features and HTTP transactions. This system is not capable of detecting the various malicious URLs that contains other malicious features. The proposed system is capable of detecting such URLs. Lawrence Kai Shih and David R. Karger[19] have proposed machine learning algorithm which analyze each links of the URLs and the visual placement of those links on a referring page to classify the URLs as either benign and malicious. The experimental results show that their approach is efficient and faster in detecting malicious web pages than those algorithms that uses content based features to detect the malicious websites. Ram Basnet et al [3] have proposed a biased support vector machine algorithm to detect phishing attacks. The algorithm analyzes various features of URL such as the ip address, number of domain, sub domain names and URL based image source to detect the phishing attacks. Fuqiang Yu [8] proposed an BM (Boyer-Moore) pattern matching method for URL classification. This method compares the URL source code with the virus characteristics to detect malicious URLs. It is not capable to detect various emerging attacks.

Min-Yen Kan and Hoang Oanh Nguyen Thi [15] have proposed to machine learning approach which demonstrates the usefulness of the URLs in the web page classification. They have implemented a classifier which performs the URL segmentation and feature extraction for classification. The experimental results shows this approach is efficient than other baseline approaches. Huajan Hung et al[11] proposed a support vector machine based approach to detect phishing URL. They used 23 features to construct the vector. The experimental results prove the efficiency of the system. But the features like brand name are difficult to analyse in real time and most of the features are applicable to simulation. The features of the proposed approach are appropriate for real time applications. Romil Rawat [22] proposed a method to detect malicious URL using ZeroR, OneR and Random Forest algorithms. The web server break down [12] is a major problem in web application. The criminals are using various parameters to down the server. The proposed approach detects the malicious parameters which affects the performance of the server.

3. ARCHITECTURE OF THE PROPOSED SYSTEM

An overview of the proposed system is depicted in figure 1. The major components are Browser, Anomaly detector, Profile and Filter.

Browser

Browser is used to collect the URL entered as input. The URL is initially compared with the black listed profile of the system. If the URL matches with profile, the browser prevents the URL from further processing and warns the user. Otherwise URL is transferred to anomaly detector for analysis. This mechanism prevents the system from analysing the same URL repeatedly and enhances the performance of the client machine where it is installed.

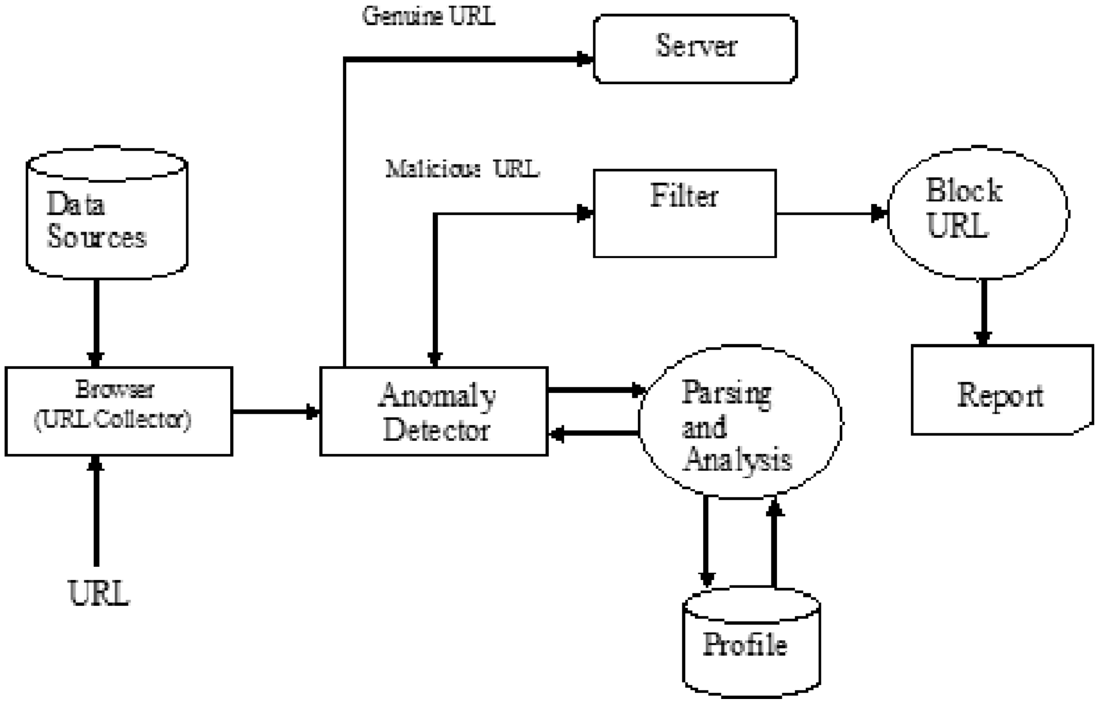

Figure 1: Architecture of the Malicious URL Detector

Anomaly Detector

It is the brain of the system that carries out the URL analysis. The Java code is used to check the URL for the occurrence of the malicious special characters and then parses the URL into three parts as domain, sub domain and path using URL Object's accessor methods in java.net package. Anomaly Detector evaluates genuineness by analyzing the various features of domain, sub domain and path and also checks the membership of the URL in the black listed profile. It enables the filter to warn the user and also blocks the malicious URL.

Profile

Profile is a structured repository. It contains a list of URLs which are blocked by the system. The dataset of profile gets updated whenever a new attack is detected.

Filter

Filter alerts the user by a warning message and blocks the malicious URLs. It also sends an acknowledgement to anomaly detector on completion of the task. The filter blocks the URL and updates the block listed profile.

4. METHODOLOGY

This paper provides a novel solution to various malicious URL attacks. For accurate detection, the proposed approach analyses various features to identify malicious URLs. The known phishing sites are used to train the system.

In this approach, the parameters like special characters, domain, sub domain and path are chosen for analysis. The proposed approach first checks the entire URLs for the occurrence of the malicious special characters. If such characters are not detected, then the system checks the genuineness of the domain. For valid domain, the system analyses the sub domain. If sub domain is a trusted one, then the system checks the path of the URL.

During the analysis if any malicious activity is detected, the system immediately blocks the URL and reports it as malicious. This system overcomes the drawbacks of the various existing approaches which are discussed in related work that detects attacks in a single dimensional manner. The features used in this approach are given below.

(A) Detecting Suspicious Special Characters

A survey was conducted to identify various special characters used for web attack. These special characters are generally not a part of genuine URLs. Occurrences of such special characters (set U) are the symptoms of the malicious attack.

U={ ‘!’,’@’,’#’,’$’,’^’,’*’,’(‘,’)’,’+’,’{‘,’}’}

These charecters are unsafe if used for URL encoding and affects the security settings and gateways. A profile is generated for the taint special characters. System compares the URL with the profile for malicious special characters. The presence of malicious special characters is reported as anomaly. For example the system blocks the malicious URL www.$google#.com which contains malicious special characters.

(B) Detecting Malevolent Domains

The malicious domain names normally resemble the trusted organization domain names with slight modification. Hence they get easily escaped from naked eyes. In the proposed system, the analyser parses the URL name as domain, sub domain and path. The system first compares the domain of the input URL with the set of malicious features. These features are extracted by lexical scanning of URL string.

The following features are used to analyse the malicious domain

Host Information

The host information helps to identify the location from where the website is hosted and the owner of the domain.

Mostly an individual is the owner for a set of malicious domains. The ownership is considered as significant feature.

IP Address

The IP address of the domain is analysed to detect the malicious domain. Several organizations provide the list of malicious IP address.

Geographical Location

It refers to the geography of the suspicious hosts IP prefix of the service provider and top level domain (TLD) gives appropriate geographical location.

Lifespan

The domains used for malicious attacks have short lifespan. The date of creation of the domain in the domain record is analysed. The domain with short life span is considered as malicious one.

Domain Name

The genuine domains name always has a meaningful English name. But most of malicious domain name are not meaningful. The Markova chain model [25] is used to analyse the text sequence of the domain name. The other properties like domain length, the number of letters and digits also analysed.

Membership the Block Listed Database

In addition to the above features the domain is compared with the block listed databases [16, 17, 18, 20].

If the domain is member of the block listed profile then it is declared as malicious.

(C) Detecting Malicious Sub Domains

After validating the legality of the domain, the system checks the sub domain of the URLs. In current scenario, the attacker simulates fake sub domains for legal domains. This is justified by www.pcrisk.com.

According to their report, the attackers of malicious programs use sub-domain services of the register domain names. The proposed system analyses the various features such as hosting account of the attacker, suspicious top level domain (TLD), graphical locations, sub domain name (like password, account info etc., are used by the attacker) and membership in the blacklisted profile.

The malicious sub domains are collected from sources like www.unmaskparasites.com. The number of malicious sub domains detected by our system is reported in Table4.

(D) Detecting Malicious Paths

After ensuring the trustworthiness of the domain and sub domain, the system analyses the path of the URLs.

The presence of one or combination of the following features in the URL is considered as malicious attack.

Parameters

The parameters listed in Table1 are used for web attack for example the CACHEDIR is used to access the list of directories in the server. CACHEDOCS is used to view the documents stored in the server.

The attacker can use CLUSTERCONFIG to access configuration setting of the web server. The parameters can be used in secured transactions, if they are detected in the path , then it is considered as malicious attack.

Table 1
Parameters

S. No.	*Parameters*
1	CACHEDIR
2	CACHEDOCS
3	CLUSTERCONFIG
4	Web config
5	IDENTIFIER
6	INITENGINE
7	LOGOPTION
8	PERSISTFILE
9	SECURITY
10	SECURITYTNSNAME
11	SUCCNOTEFILE
12	Log File Directory
13	PASSWORD

Path Tokens

The path tokens listed in the Table2 can be used by the web administrator for security purpose. Presence of such tokens in the path of a URL is used for malicious purpose. Hence occurrences of such tokens are considered as anomaly.

Table 2
Path Tokens

S No.	*Path Tokens*
1	Account
2	Webs
3	Login
4	Signin
5	Banking
6	conûrm
7	Secur
8	http

Path Characters

The malicious path characters listed in the Table3 are used to bypass the security of the web application. If such characters are present in the path then the URL is considered as anomaly.

Table 3
Special Characters

S. No.	*Malicious path characters*	*Description*
1	“../”	To bypass security filter
2	..%u2216	To bypass security Filter
3	“%2e%2e%2f”	To bypass security filter
4	“..%255c	To bypass security filter
5	%00"	To bypass rudimentary file extension checks

Top level domain

Top level domain is an added feature. Occurrence of suspicious top level domain in path token leads to malicious attack. Some of the suspicious top Level domains are given in the Table 4.

Table 4
Top Level Domain

S. No.	*TLD*	*Description*
1	Vn	Vieatnam
2	Info	Information
3	Cm	Cameroon
4	Ng	Nigeria

5. ALGORITHM

The proposed algorithm contains two phases. The primary phase analyses the given URL for the occurrence of the malicious features. ID3 algorithm is used for this purpose. The preliminary phase checks the membership of URL with the black listed profile of the proposed system, if match is found, the URL is reported as malicious.

Enhanced ID3 Decision Tree Algorithm

The enhanced ID3 algorithm initially compares the input URL with the black listed profile. If match is found it reports the URL is malicious and stops the process, otherwise it analyses the features to detect the nature of URL. This process improves the efficiency of the algorithm by avoiding repeated detection of same malicious URL.

The proposed algorithm constructs the decision tree using the concept of information gain. The output attribute represent whether the URL is “genuine” or “malicious”. The malicious features discussed in this paper are the input features. Each non-leaf node of a decision tree corresponds to an input attribute, and each arc to a possible value of that attribute. The leaf node corresponds to the value of an output attribute. The leaf node is called decision node. If the data matches with the malicious feature then the URL is classified as malicious.

The data is represented by the following form.

$$(F, C) = (f_1, f_2, f_3 \ldots f_n, C) \tag{1}$$

The vector F is the composed set of malicious features $f_1, f_2, f_3 \ldots f_n$ and C is the dependent(output) variable represents the classifications..

Initially root is tested for f1. A test instance x is first tested against the root, if the feature occurs the detector takes the left branch, otherwise it follows the right branch. The traversal from the root to leaf based on the values of the features classifies the URL**.** Traversal occurs in any left sub tree of the tree always represents the URL is malicious, further process in the left sub tree helps to identify the other malicious features present. For genuine URL the tree is a right skewed tree.

The proposed algorithm is tested against various data sources[16,17,18,20],The entropy is calculated for the given URL. It characterizes the unwanted attributes in a random collection.

The entropy values are 0 when all samples are positive or all examples are negative. If the sample of data is equally divided in terms of result, then entropy is one. The range of entropy is 0 (“perfectly classified”) to 1 (“totally random”). If target attribute takes on r different values then entropy S relative to this r-wise classification is defined as:

$$\text{Entropy}(S) = p_i \log_2 p_i \tag{2}$$

After knowing the values of the features F, the Information gain is calculated.

$$Gain(S, F) = Entropy(S) - \sum_{values(F)} \frac{S_V}{S} Entropy(S) \tag{3}$$

where S_V is the subset of the URL feature set S having value V for the feature F. Entropy of each resulting subset is weighted by its relative value. Information gain is calculated for all features. For each internal node the feature with large information gain is selected. Each leaf is assigned to one class signifying appropriate target value which classifies the URL as genuine or malicious.

The following algorithm is used to detect the malicious URLs.

```
Input: Set of Features F (Input attributes), Data set D
Output: The classification value C (output) to decide the given URL is either genuine or malicious
Compare the URL with the blacklisted profile
If ("URL belongs to the profile") then
Report URL is malicious
Return
Else
BEGIN
    If (D is NULL)
    return node N = "NO DATA"
    If(The values of D are homogenous (Malicious or genuine)
    return node N="Homogenous Value" (URL is Malicious or genuine)
    if(F is NULL) // The predicting attribute is empty,
    return node N= "Return single node with most frequency value of C in D"
    Calculate Information Gain for all features F relative to D
    Let L is attribute with max_Gain (L,D) of the attributes in F
    Let {L_i |i=1, 2......n} are the values of L
    Let{D_i |i=1,2,.....n} is the subsets of D, when D is partitioned according to the value of L.
    return a tree with root node labelled L and
    edges are labelled L_1, L_2...........L_n where edges goes to the trees
    ID3 (F_{L}, C, D1), ID3(F_{L},C,D2), ID3(F_{L},C,D3)
    If C is "Malicious URL" block the URL and store it in the black listed profile
```

The feature with the largest information gain is the root node. The internal nodes correspond to the other features of the URL. The edges separate the nodes based on the values of the features. Each leaf node represents the classification class of the URL (Genuine or Malicious). The traversal from the root of the tree based on values of the features leads to the predicted class.

Algorithm to check membership in black listed profile

The algorithm to compare the domain and sub domain with the black listed profile is given below.

Table 5
Algorithm to Compare Domain and Sub Domains with Black Listed Profile

Input : Domain and Sub domain of the given URL

Output : System blocks the malicious URL if it match with the profile.

1. Check the membership of the domain in the black listed profile D. $Y(x_1) = = \sum_{i=1}^{n} D(x_i)$

If any match is found then block the URL and report it as malicious.

2. For genuine domain, check the sub domain with the black listed profile S. $Y(x_2) = = \sum_{i=1}^{n} S(x_i)$

If any match is found, the URL is blocked and reported as malicious.

The proposed model is developed for client machine. To save the time, the proposed model contains a structured profile. The URLs blocked by the system are added in the profile. After training every input URL is initially compared with the profile. If any match is found the URL is blocked and reported as anomaly. This method helps to avoid repeated testing of the same malicious URL and supports the efficiency of the system.

Let u_k be the URL to be analysed. The set of malicious URLs are in the profile is $P(U)=\{u_1,u_2\ldots\ldots u_n\}$.

If $u_k \epsilon \{u_i\}$ where i=1,2,...n , then URL is malicious

Else if u_k is a new attack, then $\{u_i\}$ U u_k where i=1,2,...n

The algorithms are implemented in Java and run in windows 10 operating system on I3 5th Generation processor.

6. EXPERIMENTS AND RESULTS

Evaluation Metrics

To evaluate the effectiveness of the proposed approach two evaluation metrics are used. These are success ratio and error rate. Success ratio is based on the number of URLs correctly classified as malicious and genuine out of total number of URLs examined. Error rate is based on the number of URLs wrongly classified as malicious and genuine out of total number of URLs examined.

Success ratio ={(Number of URLs correctly classified as genuine + Number URLs correctly classified as Malicious) / (Total number of URLs examined)} * 100

Error rate = {(Number of URLs wrongly classified as Genuine + Number URLs wrongly classified as Malicious) / (Total number of URLs examined)} * 100

Data Set

This section describes the data sets used for evaluation. The data are selected using simple random sampling method. The sample selection is done based on the formula given below.

$$n/N \qquad (1)$$

where n is the size of the sample and N is the total population. Two data sources are used for genuine URL and four data sources are used to collect malicious URL. Each data source 100 samples are collected from first 10,000 URL using random sampling method, where n=100, N=10000 as per the formula (100/10000) 1% of sample is selected from the population. The data samples are collected from the repositories and the proposed system is trained using adequate samples to accurately classify the genuine and malicious URLs.

The genuine URLs are extracted from two data sources. The first one is DMOZ[6] open directory project. It is a directory whose entries manually verified by the editors. The second source of genuine URLs are the Random selector of Yahoo directory [26]. Altogether, 200 Genuine URLs are (100 from each data source) collected.

The malicious URLs are collected from four data sources [16, 17, 18, 20]. Yahoo Phish tank, Malcode, Malware database and Malware domain list. The user can post the malicious URLs in the data source and the nature of malicious activates are verified and added in the list.

Most of the malicious URLs listed by the Phishtank are submitted by the user and are properly verified. Malware domain list is a non-commercial community project. It can be freely used. Malware black list is the one of largest repository of malicious URLs to help researchers. Malcode is the database of domains with malicious executables. 400 malicious URLs are (100 from each data source) collected.

The proposed ID3 decision tree algorithm gathers the values of the features. The URL feature extractor is implemented as Java URL class which interacts with browser to collect the features. For every input URL, the feature extractor immediately queries features values for special characters, domain, sub domain and path.

The system analyses the values and declares the given URL is either genuine or malicious. To train this algorithm, the identified features are compared with training data. The data set consists of 600 URLs of which 200 URLs are genuine and 400 URLs are malicious. The data is split randomly, 31% of the URLs for training set and 69% as the test set. The sets are disjointed.

Table 6
Data set for training and testing

Purpose	*Genuine URLs*	*Malicious URLs*	*Total*
Training	62	124	186
Testing	138	276	414

The proposed application collects the features from the URL to update the data set as a continuous process to yield good classification results.

RESULTS

After building the data source, the proposed system is used to detect and classify the URLs. The results of the experiment are reported in Table 7. The system is accurate in detecting malicious URL under appropriate classification. Most of the malicious URLs contain malicious domains. The percentages of URLs that contains malicious special characters are reported to be very low except for the data set of phistank. A moderate number of URLs are malicious due to invalid or malicious paths.

Table 7
Categories Malicious URLs Detected in various Data sources.

Category	*Phistank*	*Malware black list*	*Malcode Database*	*Malware domain list*
Malicious Domains	30	35	37	41
Malicious Sub domains	21	27	28	22
Malicious path	18	22	20	9
Malicious special characters	20	3	2	5

Evaluation of the System

The result analysis of our system is shown in the Table 8. The success ratio is 89 % in Phistank[20] 8 % in Malware blacklist[17] , 87 % in Malcode database[16] and 77 in Malware domain list[18]. The success ratio for genuine URLs are 93% in DMOZ[6] and 95% in yahoo directory[26].

The analysis of the detected malicious URLs in each category is shown in figure 2. 42.06 % of the URLs have malicious domains, 28.82%of the URLs contain malicious sub domains. The URLs reported as malicious due to malicious paths are 20.29 % and that contain malicious special symbols are 8.82%.

Based on the literature three well known classification algorithms are chosen to compare the performance of the proposed algorithm. They are lexical feature based phishing URL detection method [1] that use confidence weighted algorithm, URL attack detection[22] using Support Vector Machine, and Malicious URL Detection Algorithm based on BM Pattern Matching[8]**.** The data is split into two folds for training and testing. For

Table 8
Result Analysis of the Proposed Approach

Efficiency	*Data sources*					
	Malicious URLs				*Genuine URLs*	
	Phistank	*Malarkey black list*	*Malcode Database*	*Malware domain list*	*DMOZ*	*Yahoo directory*
Number of URLs in the Data Sources	100	100	100	100	100	100
Success Ratio	89%	87%	87%	77%	93%	95%
Error Rate	11%	13%	13	23%	7%	5%

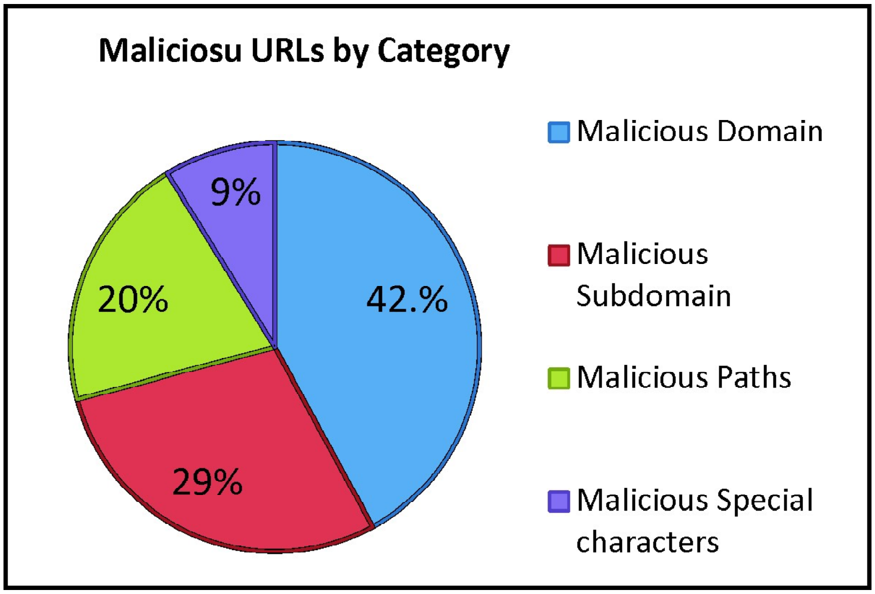

Figure 2: Analysis of Percentage of Malicious URLs by Category

comparison, the algorithm is executed in test fold. In proposed approach the decision tree algorithm is used with all derived features. This experiment is conducted to check the capability of the proposed system in detecting new types of attacks. For this purpose the URLs similar to the training data sets are removed from the test datasets. The results show that our proposed system is capable of detecting new attacks. Table 9 compares success ratio of the proposed system in detecting malicious URLs with other classifiers.

Table 9
Efficiency of the Proposed System

Data Source	*Proposed Enhanced ID3 Algorithm*	*Confidence Weight Classification*	*BM Pattern Matching*	*SVM*
Phish Tank	89	81.21	83.21	76.21
Malware black list	87	86.22	85.07	74.23
Malcode database	87	84.5	79.60	73.11
Malware domain list	77	78.14	72.11	61.42

Table 10 shows the error rate (False positive and False Negative) of our classifier in detecting malicious URLs. Our novel approach has efficiency in all data sources except for the malware domain list. The proposed system reports low error rate.

Table 10
Error Rate Analysis

Data source	*Proposed Enhanced ID3 Algorithm*	*Confidence Weight Classification*	*BM Pattern Matching*	*SVM*
Phish Tank	11	18.79	23.79	15.89
Malware black list	13	13.78	14.93	25.77
Malcode database	13	15.5	20.40	26.89
Malware domain list	23	21.86	27.89	37.58

7. CONCLUSION

In this paper, we propose a methodology to detect malicious URLs based on ID3 decision tree algorithm. The novel approach provides a unique solution to identify malicious URLs under various clauses like special characters, domain, sub domain, and path. The overall success ratio of 84.56% shows the efficiency of the system. In future this will be extended to address the emerging threats and dynamic approaches of the attacker in this area.

REFERENCES

[1] Blum, A, Wardman, B, Solorio, T and Warner, G. “Lexical feature based phishing URL detection using online learning”, In Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security, pp. 54-60, 2010.

[2] J.P. Anderson, “Computer Security Technology Planning Study”, Anderson James P and Co Fort Washington PA, Vol. 2, 1972.

[3] R. Basnet, S. Mukkamala and A.H. Sung, “Detection of phishing attacks: A machine learning approach”, In Soft Computing Applications in Industry. Springer Berlin Heidelberg, pp. 373-383, 2008.

[4] H. Trichili, M.S. Bouhlel and L. Kamoun, “A review of watermarking techniques: applications, properties, and domains”, Journal of testing and evaluation, Vol. 31, No. 4, pp. 1-4, 2003.

[5] W.S. Chou, “New Algorithm of Similarity Measures for Pattern-Recognition Problems”, Journal of Testing and Evaluation, Vol. 44, No. 4, pp. 1473-1484, 2015.

[6] DMOZ database for genuine URLs available at www.dmoz.org

[7] F. Ricca and P. Tonella, “Detecting anomaly and failure in web applications”, IEEE Multi Media, Vol. 13, No. 2, pp. 44-51, 2006.

[8] F. Yu, “Malicious URL Detection Algorithm based on BM Pattern Matching”, International Journal of Security and Its Applications, Vol. 9, No. 9, pp. 33-44, 2015.

[9] D.J. Guan, C.M. Chen and J.B. Lin, “Anomaly based malicious url detection in instant messaging”, In Proceedings of the joint workshop on information security (JWIS), 2009.

[10] G. Xiang and J.I. Hong, “A hybrid phish detection approach by identity discovery and keywords retrieval”, In Proceedings of the 18th international conference on World wide web, pp. 571-580, 2009.

[11] H. Huang, L. Qian and Y. Wang, “A SVM-based technique to detect phishing URLs”, Information Technology Journal, Vol. 11, No. 7, pp. 921, 2012.

[12] H. Huang, T. Wang and J. Ke, “Random Policy for an Unreliable Server System With Delaying Repair and Setup Time Under Bernoulli Vacation Schedule”, Journal of Testing and Evaluation, Vol. 44, No. 3, Pp. 1400-1408, 2016.

[13] H. Choi, B.B. Zhu and H. Lee, "Detecting Malicious Web Links and Identifying Their Attack Types", Web Apps, Vol. 11, Pp. 11-11, 2011.

[14] J.P. Magalhaes and L.M. Silva, "Detection of performance anomalies in web-based applications", In 9th IEEE International Symposium on Network Computing and Applications, pp. 60-67, 2010.

[15] M.Y. Kan and H.O.N. Thi, "Fast webpage classification using URL features", In Proceedings of the 14th ACM international conference on Information and knowledge management, pp. 325-326, 2005.

[16] MALECODE. Database of malicious domains/IPs hosting executables available at http://malcOde.com

[17] MALEWARE BLACK LIST. Free Malware URLs updated in real-time available at

[18] *http://www.malwareblacklist.com.*

[19] MALEWARE DOMAIN LIST Maleware domain list available at http://www.malwaredomainlist.com/mdl.php.

[20] L.K. Shih and D.R. Karger, "Using urls and table layout for web classification tasks", In 13th international conference on Proceedings of the World Wide Web, pp. 193-202, 2004.

[21] PHISHTANK. Open DNS project available at http://www.phishtank.com/ phish_archive. php.

[22] RSAAnti-Fraudat http://www.rsa.com/solutions/consumer_authentication/intelreport/ 10763_Online_Fraud _report_0210.pdf

[23] Romil Rawat, Megha Zodape, Praveen kataria, chandrapal singh dangi "URLAD (URL attack detection) - using SVM" , International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, No. 1, 2012, ISSN2277:128x.113-120.

[24] C. Seifert, I. Welch, P. Komisarczuk, C.U. Aval and B. Endicott-Popovsky, "Identification of malicious web pages through analysis of underlying DNS and web server relationships", In LCN, pp. 935-941, 2008.

[25] Y. Pan and X. Ding, "Anomaly based web phishing page detection", In 22nd Annual Computer Security Applications Conference, pp. 381-392 , 2006.

[26] Y. He, Z. Zhong, S. Krasser and Y. Tang, "Mining dns for malicious domain registrations", In 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), pp. 1-6, 2010.

[27] Yahoo Inc. Yahoo Random URL Generator. http://random.yahoo. com/bin/yrl/, October 2011.