# Detecting Website Defacement using Machine Learning Techniques

M SwarnaSudha[1], Aishwarya K[2], Sujitha S[2], and Susmitha M[2]

[1]Assistant Professor, Department of Computer Science and Engineering, Ramco Institute of Technology,Tamilnadu , India.
[2]UG Students, Department of Computer Science and Engineering, Ramco Institute of Technology, Tamilnadu , India.

Email id: swarna@ritrjpm.ac.in[1]

*Article Received: 1May 2021* *Article Accepted: 14 May 2021* *Article Published: 29 June 2021*

Citation
SwarnaSudha, M., Aishwarya, K., Sujitha, S., & Susmitha, M. (2021). Detecting Website Defacement using Machine Learning Techniques. *Journal of Next Generation Technology*, 1(1), 47-55.

## Abstract

Spam-advertised e-commerce, money theft, and malware propagation are just a few of the illicit activities made possible by the internet. The clear reality that unsuspecting people access their pages is the universal denominator, regardless of the particular motivations for these operations. These visits can also come from email, online search results, and links from other websites. In most cases, however, the consumer must take action, such as clicking on a specified Uniform Resource Locator (URL) (URL). It is vital to recognize and respond to dangers as quickly as feasible. Blacklists have traditionally been utilized to finish this identification process. Blacklists, on the other hand, are limited in scope and are unable to detect newly established defaced URLs. Machine learning methods have gotten a lot of interest in recent years as a way to improve the generality of defaced URL detectors. Many vandalized pages are never blacklisted, presumably because they are too new or were never or incorrectly reviewed. We study the output of many well-known classifiers in detecting defaced URLs as a binary classification issue, including Nave Bayes, Multi-layer Perceptron, Decision Trees, Random Forest, and k-Nearest Neighbors. We used 2.4 million URLs (instances) and three million functions from a publicly available data set. The bulk classification approaches achieve appropriate prediction rates without necessitating either specialized feature selection strategies or the assistance of an internet site professional, according to the computational simulations. Random Forest and Multi-layer Perceptron, in particular, achieve the highest accuracy.

***Keywords:*** *World Wide Web (WWW), Data Mining, Defaced Webpage Detection, URL dataset, Random Forest, KNN, ROC.*

## I. Introduction

Over the last several years, the amount and variety of online resources, such as social networking, websites, and video sharing pages, have exploded. New means for residents to connect, as well as new possibilities for other criminals, are generated as a result of these activities. The fact that Google identifies over 300,000 vandalised websites per month could be a clear indicator and proof that criminals are exploiting these gaps. With over half of the world's population having access to the Internet, unscrupulous actors have turned to the web to transmit tampered material. For the normal individual, the flexibility and accessibility of making content instantaneously accessible through the web is ideal, but it's also ideal for a malevolent intruder. To begin, an intruder in charge of the malware hosting site will quickly modify the content that the URL leads to until the payload is discovered as defaced. To avoid detection or make it more difficult for security experts, bad actors frequently offer different defaced information based on the condition in which people visit the URL from the network. Malware developers frequently employ server-side polymorphism as well. Each time a URL is visited, a new piece of content is created and transmitted to the unfortunate victim.

The Uniform Resource Locator (URL) is a global address for information and other services on the World Wide Web. The two primary components of a URL are the protocol identifier, which indicates the protocol to use, and the resource name, which specifies the IP address or name where the resource can be accessed. The protocol identifier and hence the resource names are separated by a colon (:) and two forward slashes (/). In actuality, one-third of all websites are thought to be potentially defaced, showing that defaced URLs are widely used to perpetrate cybercrime.

A defaced URL or website may host a variety of unsolicited material, such as spam, phishing, or drive-by vulnerabilities, to launch assaults. Unscrupulous people access such websites and become victims of a variety of fraud, including money loss, personal information leakage, and malware download. Drive-by downloads, phishing, social engineering, and spam are some of the more popular threats that use defaced URLs. Drive-by-download is where malware is downloaded simply by clicking a URL. Phishing and Social Engineering attempts impersonate legitimate websites to lure people into disclosing confidential or personal details. Spam is defined as unsolicited messages sent with the goal of advertising or phishing. Assaults of this nature occur in large numbers and cause billions of dollars in damage each year. Effective techniques for quickly recognising defaced URLs would greatly benefit in the battle against a wide range of cyber-security threats. As a result of this collaboration, academics and physicians have been able to develop viable Defaced URL Detection systems. The importance of understanding website functionality and behaviours in order to quickly identify and respond to threats. The phrase "website defacement" refers to an attack on a website that modifies the site's or page's exterior look. We looked at the actual functionality of a URL before attempting to reveal characteristics of defaced and non-defaced websites. We created a framework with a community of coaching data to aid the investigation. The findings of the experiment reveal that using coaching data and appropriate algorithms, the proposed algorithm can locate defaced websites.

## II. Related Work

Defaced URL identification with machine learning has been thoroughly researched in recent years. Dongjie et al [1] suggested a scheme that included three types of Online spam methods used by defaced websites: redirection spam, secret Iframe spam, and material covering spam. Then, to invalidate Site spams, he proposed a substitution prevention system that adopts users' attitudes and takes screenshots of defaced web sites. As a classification algorithm, the proposed detection approach employs a Convolutional Neural Network, which is a kind of deep neural network. Two separate trials are performed to verify the tactic's efficacy. To begin, the proposed approach was put to the test, and it was found to work with a complicated data set that had been developed. The results of a comparison of the suggested approach with representative machine learning-based detection systems are reviewed. Second, the proposed technique was employed to identify defaced websites in a real-world Network scenario over the course of three months. These results show that the proposed method outperforms the competition and may be applied in a real-world Web setting.

In [2,] Xiaodan et al suggested URL embedding (UE) as a replacement methodology for studying correlations across diverse domains, and our method is commonly used to evaluate the coefficients among these URLs. The most important condition for UE is that the URLs be represented in a dispersed manner. It focuses on a distributed representation of domains and acquires a coffee dimensional vector using a neural network. The mapping between URLs and their distributed representations must be stored for the UE model. Because the domain embedding model would hold various dimensional vectors in memory, one clear disadvantage of the strategy is the size of the space required to store it, which necessitates the use of multiple spaces. Before settling on a measurement for realms, a variety of options were considered. They claimed that defaced websites are usually considered as words, and that during this research, a distributed representation for defaced websites is frequently taught utilizing name system (DNS) searches.

The Mal JPEG method, proposed by Aviad et al. [3, is a machine learning-based approach for quickly recognizing unknown defaced JPEG images. By statically extracting 10 basic but discriminative properties from the JPEG file structure, Mal JPEG employs a machine learning classifier to discriminate between beneficent and defiled JPEG images. Because of its lossy compression, JPEG is the most widely used image format. It is used by practically everyone, from individuals to large companies, and it can be downloaded into almost any smartphone. Because of its innocuous credibility, ubiquitous use, and high potential for abuse, cyber criminals employ JPEG photos as an attack vector. Although machine learning approaches have been found to be effective in detecting known and unknown malware in a variety of domains, we are unaware of any specific machine learning methods that have been utilized to detect hacked JPEG images.

Doyen et al [4] proposed an approach that includes a complete study and systemic interpretation of Defaced URL Detection tools and leverages machine learning. On the internet, there are a variety of risks. They use a variety of methods to attack the infrastructure, including rogue websites that sell stolen goods and other forms of cybercrime. Consumers are duped into revealing personal information and their machines are infected with malware, resulting in financial theft. Researchers have proposed a number of features for identifying defaced URLs, all of which will provide useful information. The different types of features are blacklist features, URL-based Lexical Features, Host-based Features, and Content-based Features. Online research has been used to actively examine and implement Defaced URL Detection operations. Online learning is commonly classified as First Order Online Learning, Second Order Online Learning, Cost Sensitive Online Learning, and Online Active Learning. In a variety of cyber-security scenarios, detecting altered URLs is critical, and machine learning technologies are clearly a viable path. This book also includes topics about functional device architecture, transparent scientific difficulties, and future research objectives.

Anand et al [5] suggested a method titled "Detection of phishing URLs utilizing machine learning techniques," in which the author addresses the rise of phishing websites and offers techniques for extracting features and implementing machine learning algorithms to identify an alternative. They must extract features such as traffic rank information, lexical features, page rank, and so on, and they must present a comparison of different machine learning algorithms. We'll provide a comparative overview of all the algorithms and also the precision of the preferred algorithm to prove the result, rather than a hard and quick result showing the easiest algorithm isn't wiped out the article. Users who use a browser for the first time have no clue what's going on behind the scenes. Users can be duped into giving away their passwords or uploading tampered files. Their aim is to create a Chrome extension that acts as a middleman between users and, as a result, defaced websites, reducing the risk of users succumbing to them. Furthermore, it is impossible to gather all negative material and even it is subject to change. To combat this, we're using deep learning to train the tool and categories new material it encounters into groups such that appropriate actions can be taken.

Frank et al [6] proposed defaced URL detection as a binary classification issue and tested the effectiveness of several well-known classifiers, including Naive Bayes, Support Vector Machines, Multilayer Perceptron, Decision Trees, Random Forest, and k-Nearest Neighbors. A graph structure that appears like a tree, with each node indicating a test on 13 an attribute, is known as an option tree. The goal of this technique is to represent the facts while maintaining a simple model. Feature selection can be a sensitive and tough challenge in a data collection with over 2 million entries, each with over 3 million attributes, making pattern detection and correlation two computationally costly operations. The nominal aspects of a URL are coded in the dataset as a series of binary attributes, each with one of the available values. Unless one of the binary attributes has a value of 1, none of the individual attributes possess all of the knowledge about the function when a categorical value is distributed over numerous binary attributes. Calculating interdependencies between over 3 million characteristics could be a computationally demanding procedure. Due to the enormous amount of binary features, pattern recognition does not ensure a ready-to-use technique (only 67 out of three million have real-values). Instead, we used a number of function selection strategies, such as feature sets A and B, which contain both binary and real-value attributes, and feature set C, which exclusively contains real-value attributes.

Anton et al [7] proposed a method for analyzing URLs in network traffic and changing detection models to accommodate new defaced content. Researchers discovered a number of ways to counteract hazards that spread through the internet. One of them is to detect defaced material or maybe URLs related with it using signature-based identification. Blacklisting and rule-based or heuristic identification, as the authors illustrated, are all great approaches, but they all suffer from the same flaw: the inability to adapt to new defaced URLs and URL trends. As host-based functions, use IP addresses, WHOIS assets, location, properties, geographic properties, or link speed. External 14 information such as the URL's indicated resource are not taken into account by extracted features, which rely entirely on the URL string. The machine learning approach was validated using the One-Side Class (OSC) Perceptron, which has a high detection rate and 0% false positives. These algorithms are frequently employed to identify defaced URLs with a short lifespan or those used in targeted attacks, according to the research.

Adrian et al. [8] proposed a technique in which the major areas of emphasis are the employment of several machine learning algorithms and unsupervised learning approaches for detecting defaced URLs with relation to memory footprint. This study examines several machine learning methodologies and unsupervised learning methods for detecting defaced URLs without taking into account the URL's server content or any extracted external properties. Earlier solutions focused on signature-based detections to detect malware or URLs that reference defaced content. Later on, more complex structures were put in place. In terms of the beneficial impacts, the requirement to employ downloaded content can create issues when dealing with high-complexity attacks. An intruder who discovers the detection system's IP address, for example, could change the downloaded data from malicious to benign. The technologies mentioned have a low false positive rate and a modest memory footprint for an outsized dataset.

Chaitrali et al. [9] proposed a design for kayo, a system that distinguishes between dangerous and benign mobile web sites. Kayo allows for this decision to be made based on the static attributes of a web page, such as the number of iframes and the presence of known phony phone numbers. The 15 strategy first shows why mobile-specific techniques are needed, then analyses a number of current static elements that are significantly associated to mobile-defaced websites. The researchers then used kayo to accurately classify over 350,000 reported benign and defaced mobile web pages. Furthermore, it discovers, characterizes, and reports a range of pages that Google Safe Browsing and Virus Total ignore. Finally, kayo is used to construct a browser function that protects users in real time from defaced smartphone websites. As a result, it provides the primary static analysis tool for detecting defaced mobile site pages.

Mohammed et al. [10] proposed a method for classifying URLs automatically based on their lexical and host-based properties. The complete dataset is clustered, and a cluster ID (or label) is established for each URL, which is subsequently employed as a predictive function by the arrangement. To categorise URLs, online URL credibility services are used, and the categories given are used as a supplemental source of data that allows the system to rate URLs. With a precision of 93-98 percent and a low false positive rate, the classifier finds an extremely large number of phishing hosts. URL clustering, grouping, and categorization mechanisms use a conjunction to assign a level to URLs. The approaches employed by researchers to solve the challenge of phishing URL identification and classification are discussed in this article. The current project is intrinsically tied to the thesis. They achieve a classification accuracy of roughly 95% by extracting lexical and host-based information from URLs. Microsoft Reputation Services (MRS) returns one or more groups of threat levels for each URL supplied. The Severe, Moderate, and Benign bags in the current work are similar to the previous 16 list. To rate URLs, researchers utilized clustering, sorting, and categorization, with cluster labels raising classifier accuracy from 97.08 percent to 98.46 percent.

### A. Machine Learning Methods

This research uses two methods to detect defaced web pages, they are

- K-Nearest Neighbors (KNN)
- Random Forest Algorithm

K-Nearest Neighbor's (KNN) is a machine learning technique for solving regression and classification issues. KNN algorithms use prior information and a similarity score to find new data points. The letter 'K' in K-Means denotes the number of clusters the algorithm is attempting to identify/learn from the data. Because this is frequently utilized in unsupervised instruction, the clusters are frequently ambiguous. We look at the K nearest training data points for and test datum in the K means algorithm and identify the most frequently occurring groups to allocate to the test data. As a result, K represents the total of coaching data that is near to the assessment data point used to determine the category. Run the KNN algorithm numerous times with different K values to determine the K that best suits your results. Choose the K that decreases the amount of errors while maintaining the algorithm's ability to produce correct predictions. Figure 1 shows the KNN Formula.

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{1}$$

The number of clusters the algorithm is attempting to identify/learn from the data is denoted by the letter 'K' in K-Means. Let (Xi, Ci) be the data points, with I = 1, 2......., n. For each I Xi represents feature values and Ci represents labels for Xi.

Assuming that the number of classes is 'c'Ci 1, 2, 3,......, c for all I values. Let x be the degree to which the mark is unknown, and we wish to discover the label type using k-nearest neighbour techniques. Instead of looking for the most essential function when breaking a node, it seeks for the simplest feature from a random group of features.

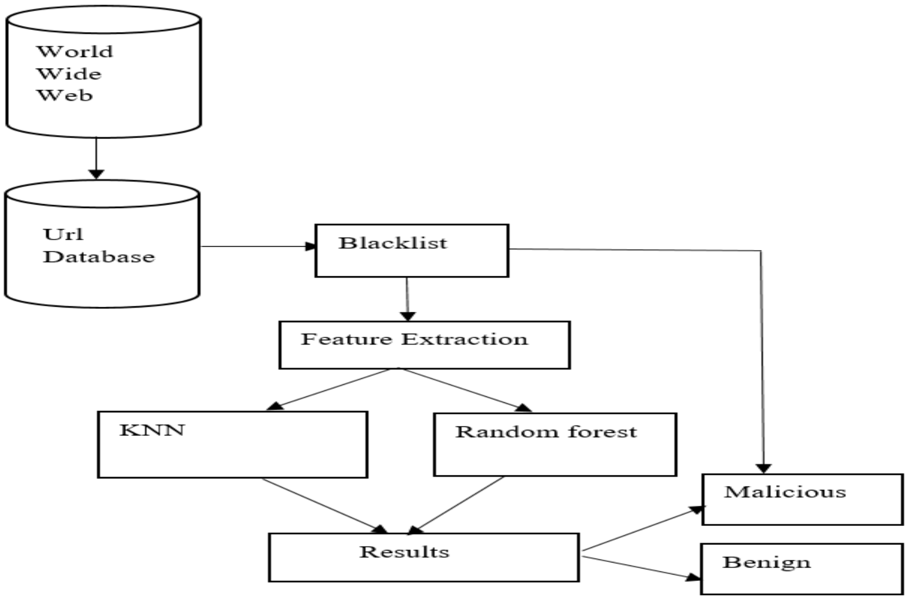


**Figure 1.** The overview of labor

Figure 1 illustrates this. Determine what information can be gleaned from current databases or sources, compile the data, and store it in a compilation log. To discover their different qualities, sets of important features are analyzed using a machine learning algorithm and trial and error. Specific data sets are cleaned and organized in a uniform style for easy access and interpretation, enabling for faster and more informed decisions. Separate a data set into a training set and a research set, with the majority of the data being used for training and a smaller portion being utilised for testing (Training data – 70% Testing data – 30%). Examine the precision and consistency of the algorithm. The degree to which the experimental value correlates to a specific quantity of material within the matrix is referred to as accuracy. Precision tests are conducted to determine how comparable human dimensions are to one another. (Confusion Matrix, ROC curve, accuracy ranking, Classification Report.)

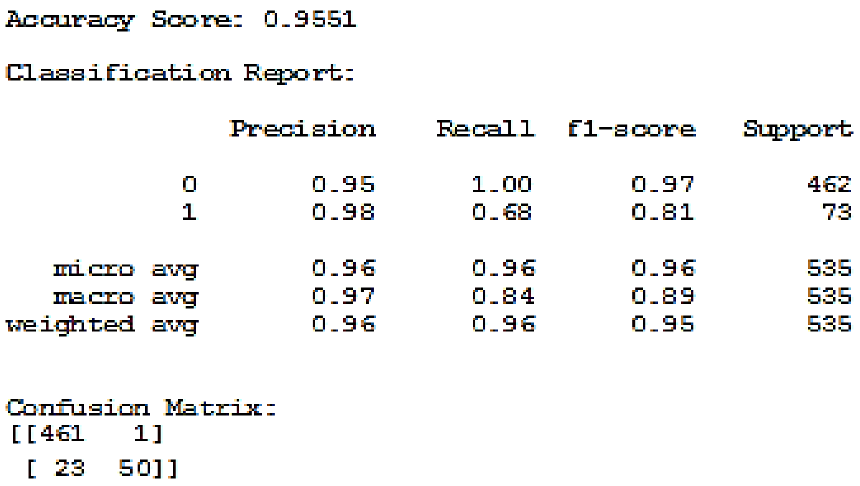
## III. Dataset Description

The goal of this project is to use machine learning to detect defaced webpages. It has a total of 15 characteristics.

1. **Url**: it's the study's anonymous identifier for the url being examined. • Url length: it's the number of characters in the url.
2. **Numberspecialcharacters**: the number of special characters in the url, such as "/," "percent," "#," "&," ".," and "="
3. **A character set** is a collection of symbols and encodings.
4. **Server:** this is a categorical value that denotes that the packet response was received by the server's operating system.
5. **Content length**: this specifies the size of the http content header.
6. **Who is country**: this variable represents the country values obtained from the server response.
7. **Who is statepro**: it's a variable that stores the values of states received from the server response.
8. **Who is regdate**: this field displays the server registration date, which is formatted as dd/mm/yyyy hh:mm.
9. **Who is updated date**: using who is, we were able to obtain the most recent update on the server.
10. **TCP conversation exchange**: the number of tcp packets exchanged between the server and the client is represented by this variable.
11. **DIST remote TCP port**: this is the number of ports discovered, which is distinct from tcp.
12. **Remote IPS**: this variable contains the total number of ip addresses that are linked to the honeypot.
13. **App bytes**: this is the number of bytes exchanged most frequently.
14. **Remote app packets**: received packets from the server
15. **App packets:** this is frequently the total amount of ip packets generated during communication between the honeypot and, as a result, the server.
16. **DNS query time**s: this is frequently the number of dns packets generated during connection between the honeypot and, as a result, the server.
17. **Type**: this is a categorical variable whose values describe the type of website being examined, with 1 indicating defaced websites and 0 indicating innocuous websites.

# IV. Results

## A. Random Forest



```
Accuracy Score: 0.9551

Classification Report:

              Precision    Recall  f1-score   Support

           0       0.95      1.00      0.97       462
           1       0.98      0.68      0.81        73

   micro avg       0.96      0.96      0.96       535
   macro avg       0.97      0.84      0.89       535
weighted avg       0.96      0.96      0.95       535


Confusion Matrix:
[[461   1]
 [ 23  50]]
```

**Figure 2.** The Result of Random Forest

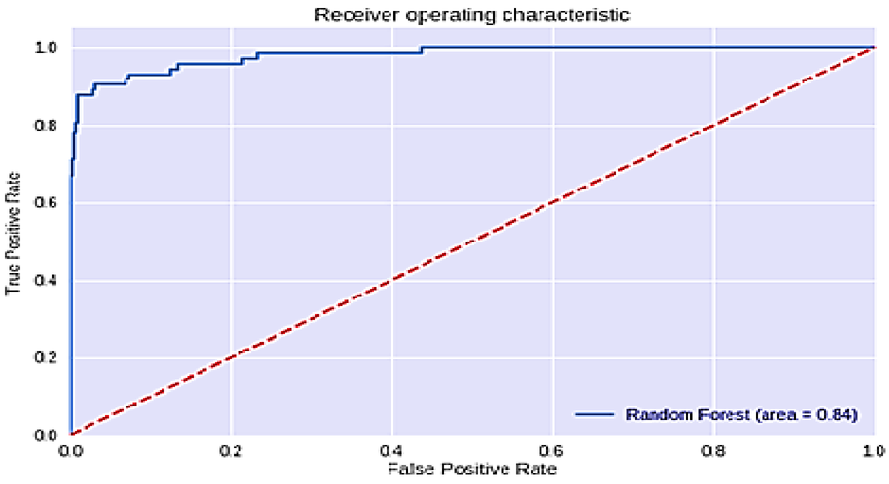
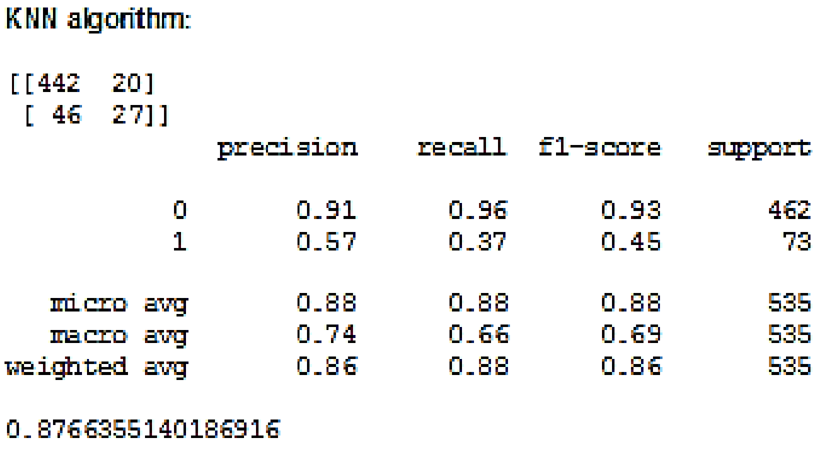


**Figure 3.** The ROC curve of Random Forest

## B. KNN Algorithm



```
KNN algorithm:

[[442  20]
 [ 46  27]]
              precision    recall  f1-score   support

           0       0.91      0.96      0.93       462
           1       0.57      0.37      0.45        73

   micro avg       0.88      0.88      0.88       535
   macro avg       0.74      0.66      0.69       535
weighted avg       0.86      0.88      0.86       535

0.8766355140186916
```

**Figure 4.** The Result of KNN

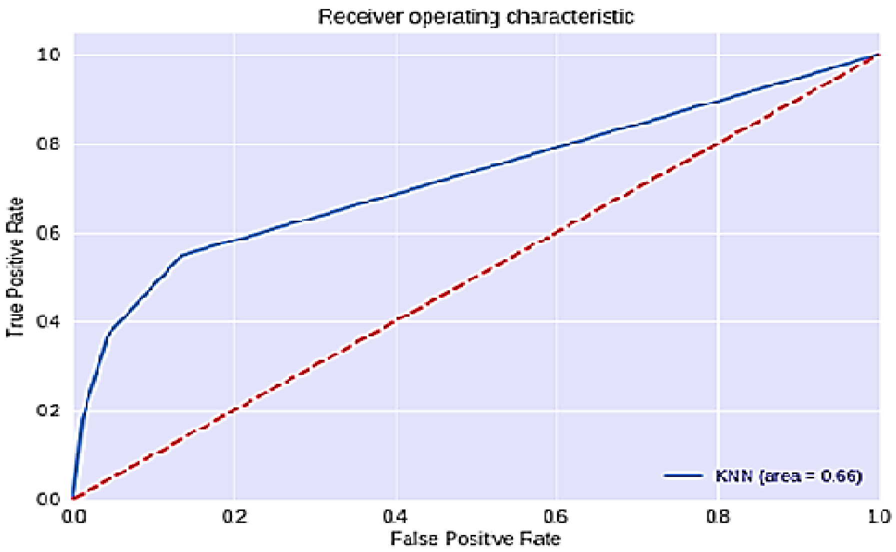


**Figure 5.** The ROC curve of KNN

## V. Conclusion and Future Work

The computer models, as well as the mathematical studies that follow, clearly show that there are significant differences between the techniques. As a result, the rating of the approaches is frequently regarded as a trustworthy and relevant ranking in terms of forecast precision. Despite the fact that both approaches achieve similar high prediction accuracy, Random Forest, followed by MLP, appears to be the most suitable classification algorithm for this situation. Random Forest likewise receives good marks, suggesting not only well-balanced and impartial prediction performance, but also confidence in the method's potential to maximize defaced URL detection while staying within permissible limitations. According to the findings of this study, when just numerical features are utilized for training, the classification techniques achieve competitive prediction accuracy values for URL classification.

Ensemble learning can be simplified function extraction and representation in the future. For managing idea drifts and other emerging problems, simpler machine learning algorithms for training predictive models are required. IT enhances the efficacy and reliability of coaching practices by combining remote learning approaches with real-time video feeds.

## References

[1] Doyen Sahoo, Chenghao Liu, and Steven C.H. Hoi, Malicious URL Detection using Machine Learning, IEEE, 2020.

[2] Xiaodan Yan, Yang XuB, Member, Baojiang Cui, Learning URL Embedding for Malicious Website Detection, IEEE, Oct 2020.

[3] Aviad Cohen, Nir Nissim, Yuval Elovici, MalJPEG: Machine Learning Based Solution for the Detection of Malicious JPEG Images, IEEE, 2020.

[4] Doyen Sahoo, Chenghao Liu, and Steven C.H. Hoi, Malicious URL Detection using Machine Learning: A Survey, IEEE, 2017.

[5] Anand Desai, Janvi Jatakia, Rohit Naik, Nataasha Raul, Malicious Web Content Detection Using Machine Leaning, IEEE, 2017.

[6] Frank Vanhoenshoven, Gonzalo Napoles, RafaelDetecting Malicious URLs using Machine Learning Techniques, IEEE, 2016.

[7] Anton Dan Gabriel, Dragos, Teodor Gavrilut, Dragos, Teodor Gavrilut, Popescu Adrian Stefan, Detecting malicious URLs: A semi supervised machine learning system approach, IEEE, 2016.

[8] Adrian Stefan Popescu, Dragos, Teodor Gavrilut, Dumitru Bogdan Prelipcean, A Study on Techniques for Proactively Identifying Malicious URLs, IEEE, 2016.

[9] Chaitrali Amrutkar, Young Seuk Kim, Patrick Traynor, Detecting Mobile Malicious Webpages in Real Time, IEEE, 2016.

[10] Mohammed Nazim Feroz, Susan Mengel, Phishing URL detection using URL Ranking, IEEE, 2015.