iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** XII **Month of publication:** December 2022

DOI: https://doi.org/10.22214/ijraset.2022.48397

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XII Dec 2022- Available at www.ijraset.com

A Machine Learning Approach for Web Application Vulnerability Detection Using Random Forest

Chandan Singh[1], V. Vijayalakshmi[2], Harsh Raj[3]
Department Of Networking and Communication, SRM Institute of Science and Technology, Chennai, India

Abstract: ***Website attacks have been one of the main threats to websites and web portals of private and public organizations. In today's digital world web applications are an important part of day-to-day life so it has become a challenging task to secure the applications. The attackers aim to extract sensitive information about the users through the URL links sent to the victims. We are trying filling the gap of traditional methods to stop the attacks, but the traditional methods fail to perform well as the attackers are becoming good at attacking the web applications. People are presently searching for reliable and consistent web application attack detection software. This model aims to secure web applications of vulnerabilities and from different types of attacks using a machine learning approach which has more accuracy compared to other machine learning algorithms since we are using Random Forest Model.***

Keywords: URL, Attacks, Extract, Web portals, Vulnerabilities, Random Forest Model.

I. INTRODUCTION

Web applications have become the most important part as it helps to reduce all complexity and makes life easy. As it has this much importance in the day-to-day life it becomes usual to some bad affects also it means it will attract lot of attentions of third parties and hackers as it handles all types of traffic like banking defense transactions or cyber bullying.[1]

Attacker can't easily take control of web applications of others for that they need an open port or a weakness of the system where they can enter through it this weakness is called as vulnerability it means nothing but the weakness using which hackers can access and perform tasks like manipulate, destroying data or changing this all can be done. [2]

Vulnerabilities are typically created unintentionally during system development. Vulnerabilities are caused by incorrect design decisions in one of the phases of the system life cycle. Bugs found and fixed during the development and testing stages are not counted as vulnerabilities, only bugs that are built into the operation of the system. If the creation is malicious and therefore intentional, the discovery and creation match. After a vulnerability is discovered, you can retroactively determine the point at which the vulnerability was created.[3]

This opens to many attacks like phishing, defacement, malware SQL injection, XSS attack and many more here we are taking only 3 types of attack they are phishing, defacement and malware.

There are many traditional techniques to detect this type of vulnerability in web applications but they are limited only some operations so there were requirements for more efficient methods to detect vulnerability so by using machine learning methods we reduce false rate and increase the accuracy. [4]

There are many algorithms which help in detecting vulnerabilities, but we need the best one which has the highest accuracy, so we are using Random Forest algorithm to detect multiple forms of attack. As we are using machine learning we need sufficient datasets to train the model for testing it. So, it becomes a major problem as datasets are not available in sufficient amounts so for some type of attack it becomes very difficult to train it.[5]

The project is used to detect the attack Phishing, Malware, Defacement attacks. And is used by accuracy of 96.6%. This project will help individuals as well as organizations in detecting attacks, which can happen while clicking on the infected link.[6]

The rest of the paper is formulated as making detailed literature study in Section II. The system tool selection, problem identifications are discussed in Section III.

The system architecture, detailed system design steps are discussed in Section IV. The implementation steps are discussed in SectionIV. The rest of the paper is concluded with future enhancement.

II. LITERATURE SURVEY

1) *Dau Hoang et al., (2018)* This paper proposes a machine learning-based method for website defacement detection. In his approach, a detection profile is automatically learned from a training dataset of both normal and corrupted web pages. Experimental results show that our method can produce high detection accuracy and false positive rate. In addition, his method does not require extensive computing resources, so it is practical for implementing an online monitoring and detection system for website defacement.
2) *Sara Althubiti et al., (2017)* In this paper, various machine learning techniques have been applied to the CSIC 2010 HTTP dataset for intrusion detection purposes. The dataset included attacks such as SQL injection, buffer overflow, information gathering, file disclosure, and so on. Experiments show that all techniques have high precision, recall and F1 rates and low FPR, except Naïve Bayes, which shows lower precision, recall and F1 rates and high FPR compared to the rest of the techniques. (Nguyen et al., 2011) extracted nine features considered important for the detection process and used the top five as selected by Weka; this provided better results, high accuracy and reduced training time.
3) *Mauro Conti et al., (2020)* Web applications are particularly challenging to analyze because of their diversity and widespread adoption of proprietary programming practices. ML is thus very useful in the web environment because it can take advantage of manually labeled data to expose human understanding of the semantics of a web application to automated analysis tools. They confirmed this claim by designing Mitch, the first ML solution for Blackbox detection of CSRF vulnerabilities, and experimentally evaluating its effectiveness. They hope that other researchers could use their methodology to detect other classes of web application vulnerabilities.
4) *Banu Diri et al., (2019)* In this paper, the author implemented a phishing detection system using seven different machine learning algorithms such as Decision Tree, Adaboost, K-star, KNN, Random Forest, SMO, and Naive Bayes, and various types of features such as NLP, word. vectors and hybrid elements. To increase the accuracy of the detection system, the key task is to construct an effective feature list. Therefore, he grouped his list of features into two different classes as NLP- based features, which are mainly human-defined features and word vectors that focus on using words in URLs without performing any additional operations.
5) *Tuong Ngoc Nguyen et al., (2019)* The paper proposed a hybrid website defacement detection model that was based on machine learning techniques and attack signatures. The machine-leaning component can detect corrupted web pages with a high level of accuracy, and the detection profile could be learned using a dataset of both normal and corrupted pages. The signature-based component helped speed up processing of common forms of spoofing attacks. The experimental results showed that the damage detection model can perform well on both static and dynamic websites, and that it has an overall detection accuracy of more than 99.26% and a false positive rate of less than 0.62%. The model is also capable of following web pages in languages other than the language of the training data web pages.
6) *Truong Son Pham et al., (2016)* In this article, we compared different machine learning techniques in web intrusion detection. In the experiments, we used the CISC 2010 HTTP dataset, which includes attacks such as SQL injection, buffer overflow, information gathering, file sharing, CRLF injection, XSS, server side include, parameter forgery, and so on. Experiments have shown that logistic regression is the best learning method for this problem. Logistic regression presents a very good performance with the highest recall as well as the highest precision. We have also tried to improve its performance using various feature extraction, feature selection and also parameter rotation techniques. The results looked better after that.
7) *Jacob Howe et al., (2018)* This paper showed that SVM, k-NN, and Random Forest can be used to create classifiers for XSS coded in JavaScript that provide high accuracy (up to 99.75%) and accuracy (up to 99.88%) when applied to a larger number of data files. data file. This shows that these classifiers can be added as a security layer either in the browser or (as intended) on the server. The training data was designed to provide a fair coverage of scripts, including scripts of different lengths and both obscured and unobscured scripts. Rather than using obfuscation as a proxy for maliciousness, data is labeled as malicious or benign. While SVM, k-NN and Random Forest were used in the experiments; other classification methods were also expected to perform well. Considering the various existing frameworks, it is clear that the Random Forest architecture is widely used.

III. SYSTEM DESIGN

A. *Problem Analysis*

1) The customary way of detecting the malicious websites is to discover and update such suspicious or malicious URLs, IPs (Internet Protocol), to the database of the websites, this method of identifying malicious websites is called as the Blacklist method.

2) To avoid from being blacklisted, attackers use a variety of and numerous ways to deceive consumers, including altering URLs to make them appear real, authentic, obfuscation, and a variety of other basic strategies such as the fast- flux. In this method, proxies are formed automatically to host the Webpages another method is to create URLs algorithmically, and so on.
3) A Heuristic based detection, it detects the attacks based on the characteristics and features that are discovered on phishing attacks, this method can also be used to detect zero-hour attacks which the Blacklist method fails to detect, but it is not that guaranteed that these characteristics always exist in the attack furthermore, bogus and nonsensical positive rate in identification is high.
4) To succeed from the downsides Security specialists are currently cantered around Artificial Intelligence methods. AI calculation needs the past information to settle on choices or expectations on the future information.

B. *Dataset*

We have taken the dataset from Kaggle which is a collection of 651191 URLs. From the dataset which we have taken 428103 are safe URLs, 96457 are defacement URLs, 94111 are phishing URL, 32520 are malware URLs and over all the websites of 651191 which is a collection of all legitimate, phishing, malware and defacement websites which can be used as a training dataset.

1) Safe: 428103
2) Defacement: 96457
3) Phishing: 94111
4) Malware: 32520

C. *Architecture*

The dataset is a combination of three dataset all together which contains all the types of attack present in the data set which phishing malware and defacement the dataset is processed and made through the feature extraction process which is a technique used to reduce the number of features in a dataset by creating a new feature set from a particular feature in the dataset. This is used when the dataset contains many characteristics, such as, makes the model difficult to fit to the data set where in feature fd_length, hostname_length, count_dir, url_length, count, count_letters, tld_length, count, count-www, count = and count percentage. These features are extracted on the basis of which malicious and legitimate URLs are differentiated as the URLs which have attack show different features compared to the legitimate website when it is found that it is legitimate website we pass it through the Random Forest model, this model then divides the website URLs on the basis of the attack into safe, malware, defacement, phishing. The detection report is then generated according to the observation and warning dialogue box with the given accuracy to the user.

D. *Design*

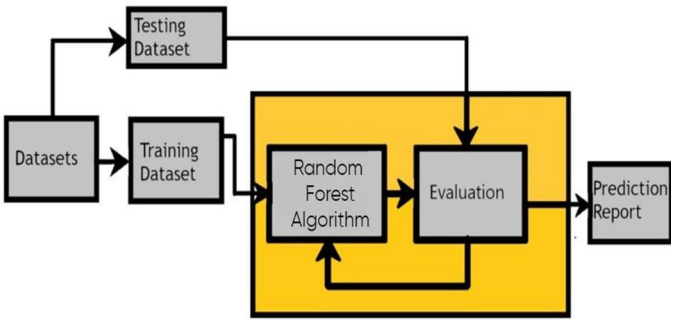


Fig. 1. Shows the Design of the proposed model.

IV. METHODOLOGY

A. *System Architecture*

The system architecture provides an overview of how the system works. Here's how this system works:
Dataset collection is the collection of data, including URLs and websites which can be either malicious or legitimate. Through the process of the feature extraction, we extract and differentiate the attacks and process them further to know whether they are legitimate or not.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue XII Dec 2022- Available at www.ijraset.com

First we insert the trained dataset with the input URL's which is further completed by checking whether the given URL is malicious or not after finding it malicious in which case it is either phishing, malware, defacement we intercept it with algorithm and show warning dialogue with the type of attack which is there if it is not a malicious website then we show the dialogue and load the page in normal manner.

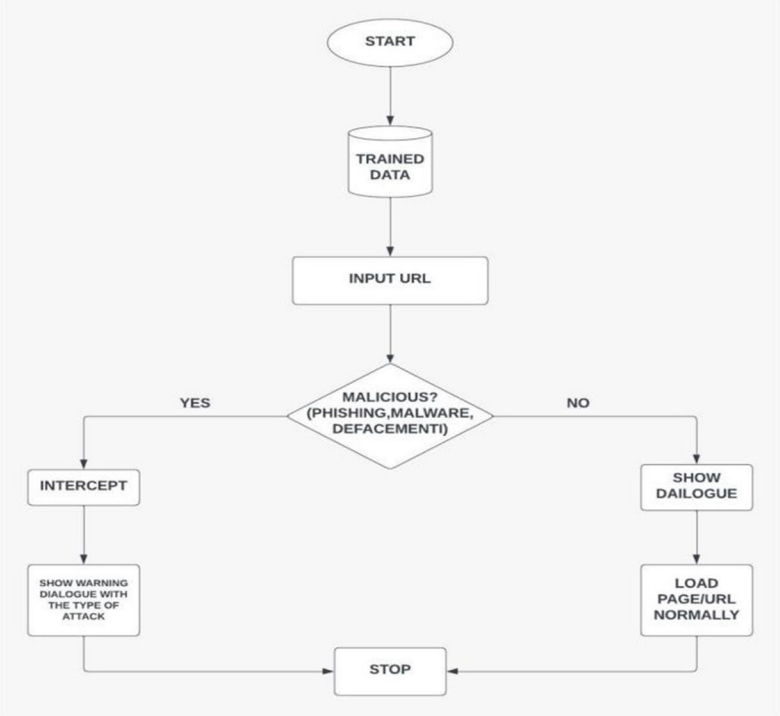


Fig. 2. Shows the Flowchart of proposed model.

B. Supervised Machine Learning

In the simplest sense, supervised learning means learning that an algorithm maps an input to a particular output. If the mapping is correct, this indicates that the algorithm has been learned correctly. If not, make the necessary changes to the algorithm so that it can be learned properly. Supervised learning algorithms can predict invisible data that will be received in the future.

The supervised learning model is used to build and improve several business applications, including:

1) *Predictive Analytics:* Today, predictive analytics is growing exponentially with the proliferation of cryptocurrency and equity trading use cases. This allows organizations to verify specific outcomes based on specific output variables, helping executives to justify decisions that benefit the organization.
2) *Analyzing Customer Sentiments:* Using supervised machine learning algorithms, businesses gain critical information such as context, emotion and intent. This data helps you better understand how you interact with your customers and improve your brand's growth.
3) *Detecting Spam:* Spam detection is a typical use case for a supervised learning model. Using supervised classification algorithms, organizations can train their databases to recognize new data patterns and anomalies and effectively organize unsolicited and non-spam communications.

C. Random Forest model

Random forest is a non-parametric (no assumption on the probability distribution of the data points) supervised machine learning algorithm. This is an extension of the machine learning classifier that includes bagging to improve decision tree performance. It combines tree predictors, and the tree relies on an independently sampled random vector. It belongs to a class of ensemble methods as it tries to reduce variance and produces an "average" decision rule from a set (the forest) of many different decision trees. These trees are constructed in such a way that when the prediction on a new data point is given by some part of the forest, it should be like the average rules produced by other parts.

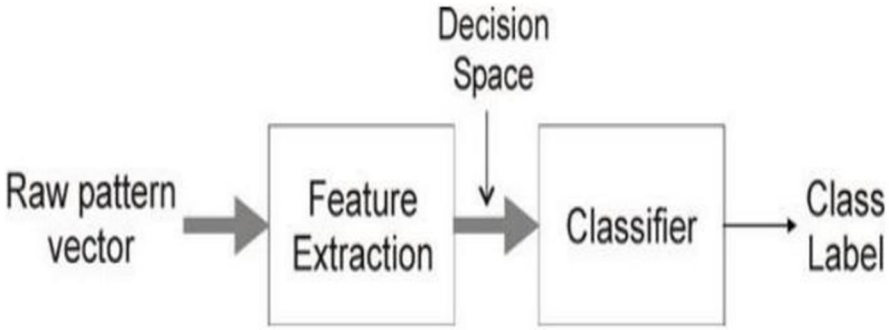


Fig 3. Random Forest architecture

1) The idea is that in many cases it is not necessary or desirable to use complex parametric models if you can get satisfactory results using much simpler non- parametric models. The simplest classifier rule in unsupervised approach is the decision boundary between classes. This would mean that we must classify all data points using a single separator line (hyper plane). The problem is that this simple rule does work well on non-linear problems. In fact, for this kind of dataset even multiple hyper planes cannot always be classified perfectly without any mistakes.
2) This is where the random forest algorithm comes into play. It produces a set of many decision trees which all give an average classification rule using different rules for each data point.
3) It can also maintain accuracy even if a large amount of information is missing. It requires a certain amount of investment compared to other algorithms, and it accurately predicts the yield of large data sets that are executed productively.

V. IMPLEMENTATION

For setting up the model to train we need to import the python packages such as Pandas, NumPy, Scikit-learn, Matplotlib, Flask. So, for importing we need to set up all package to import.

A. *Feature Pattern*

In the given model to detect attacks and classify them we need features for our database we have chosen some important features so before classifying the data we need to check if there any matching patterns that can determine the types of links on the basis of the data collected from the database.

B. *Suspicious Words*

In the current dataset which have been collected we search for the suspicious words which help to identify the treats more accurately.

C. *Pre-processing*

In the dataset we have collected for different types of attack like normal, phishing and defacement so we need to get the total information about the number of data we have on the attacks to train the model for detection.

D. *Feature Extractor*

For the model to detect we need feature on which the type of URL will be decided so for the feature will play an important role as they can improve reduction in false rate as we are going with the Random Forest algorithm, we need features for different types as every type of attack has different types of feature so we have collected the most used features in all 3 types of attack.

E. *Confusion Matrix*

The confusion matrix is a technique for summarizing the performance of a classification algorithm. If each class has an unequal number of observations, or if the dataset has more than one class, the accuracy of the classification alone can be misleading.

VI. RESULTS AND DISCUSSIONS

A. *Attack Detection*

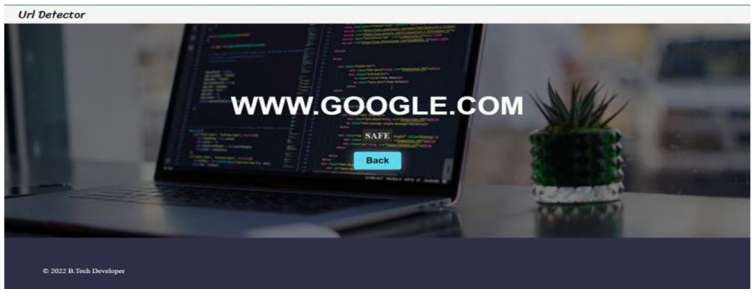


Fig 4. Safe Website

Fig. 4 shows that This URL does not contain any malicious attacks and is safe to browse.

B. *Setup*

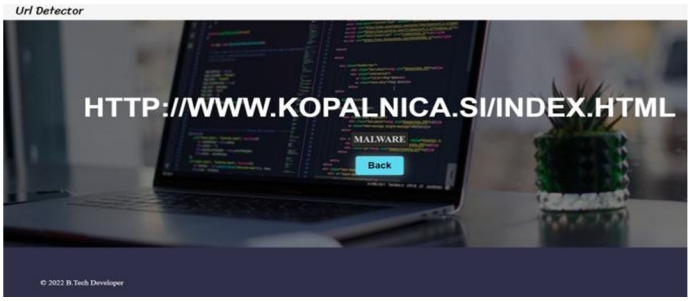


Fig 5. Detecting website with malware attack

Fig 5. shows a malicious website which consists of malware attack.

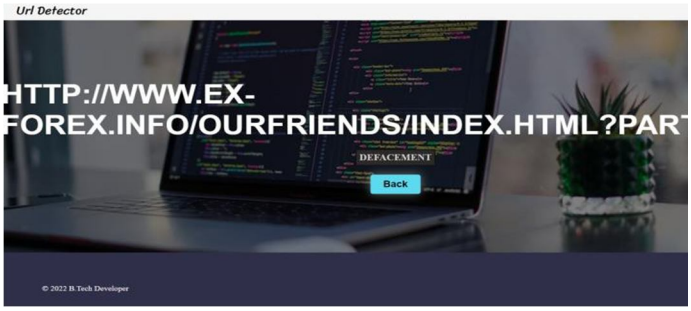


Fig. 6. Detecting website with defacement attack

Fig 6. shows a malicious website which consists of defacement attack.

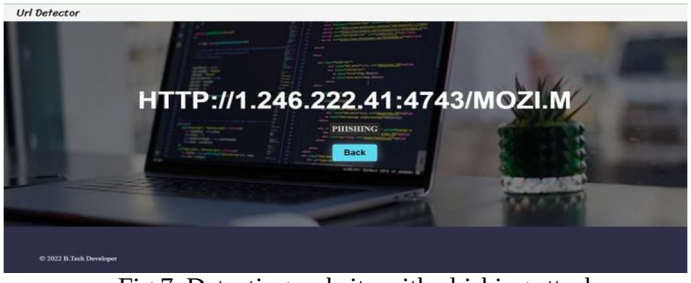


Fig 7. Detecting website with phishing attack

Fig 7. shows a Phishing website which consists of Phishing attack.

VII. CHALLENGES

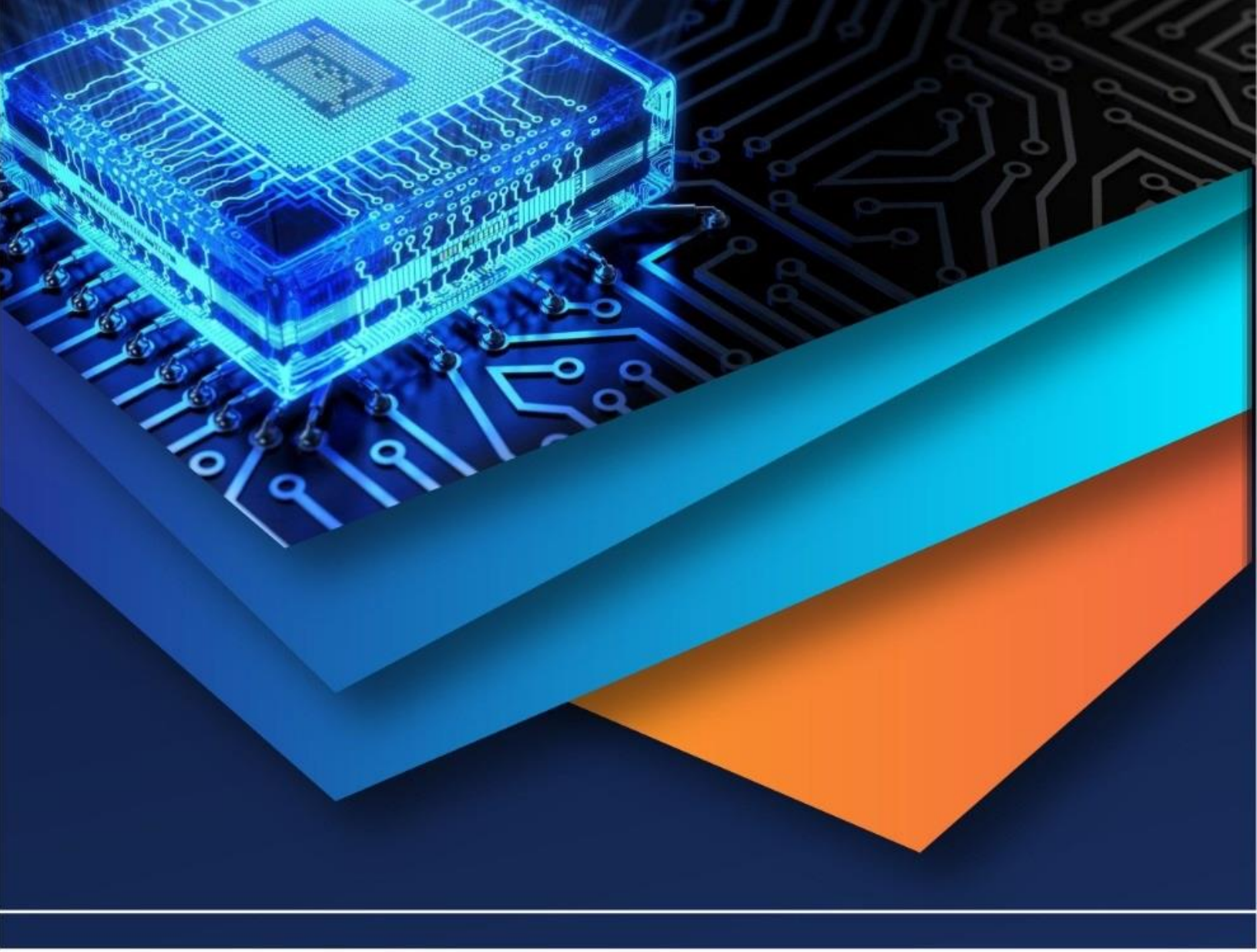
The major challenge persisting with the proposed model is the usage of large datasets. To assess the machine learning techniques, we have utilized a dataset which contains over 651191 URL's both legitimate and malicious. Each which comprises of 21 features. Each URLs has a standard. If the standard is met, it will be considered as malicious URL. If not, it will be considered as legitimate URL. So, we found that if we are using SVM and logistic regression over Random Forest then the system performance is being degraded.

VIII. CONCLUSION

As in today's internet world web technology is at its higher growth potential. The safety of the website and protection from attacks like phishing malware defacement attacks and be detected and prevented. This project creates a model which helps in the safety and security of the given website. This project is optimized at a level where we can identify the malicious and non-malicious website using datasets which and then with the procedure of feature extraction the website is noticed and then detected using Machine Learning by use of Random Forest algorithm and accuracy of the attack is determined. This model will give a safe and secure way to access website without worrying about the unfamiliarity and unpredictable behavior it possesses as we get the idea of it by machine learning it is the technology which can has impact in all the domains and in this digital world this is one of the best tools for safe and secure internet platform.

REFERENCES

[1] Hoang, X. D. (2018, December). A website defacement detection method based on machine learning techniques. In Proceedings of the Ninth International Symposium on Information and Communication Technology (pp. 443-448).
[2] Hoang, X. D., & Nguyen, N. T. (2019). Detecting website defacements based on machine learning techniques and attack signatures. Computers, 8(2), 35.
[3] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, 345-357.
[4] Jain, A. K., & Gupta, B. B. (2018). Towards detection of phishing websites on client-side using machine learning based approach. Telecommunication Systems, 68(4), 687-700.
[5] Althubiti, S., Yuan, X., & Esterline, A. (2017). Analyzing HTTP requests for web intrusion detection.
[6] Mereani, F. A., & Howe, J. M. (2018, February). Detecting cross-site scripting attacks using machine learning. In International conference on advanced machine learning technologies and applications (pp. 200-210). Springer, Cham.
[7] Pham, T. S., Hoang, T. H., & Van Canh, V. (2016, October). Machine learning techniques for web intrusion detection—A comparison. In 2016 Eighth International Conference on Knowledge and Systems Engineering (KSE) (pp. 291-297). IEEE.
[8] Calzavara, S., Conti, M., Focardi, R., Rabitti, A., & Tolomei, G. (2020). Machine learning for web vulnerability detection: the case of cross-site request forgery. IEEE Security & Privacy, 18(3), 8-16.
[9] Zolanvari, M., Teixeira, M. A., Gupta, L., Khan, K. M., & Jain, R. (2019). Machine learning-based network vulnerability analysis of industrial Internet of Things. IEEE Internet of Things Journal, 6(4), 6822-6834.
[10] Jain, A. K., & Gupta, B. B. (2018). Detection of phishing attacks in financial and e-banking websites using link and visual similarity relation. International Journal of Information and Computer Security, 10(4), 398-417.
[11] Romagna, M., & van den Hout, N. J. (2017, October). Hacktivism and website defacement: motivations, capabilities and potential threats. In 27th virus bulletin international conference (Vol. 1, pp. 1-10).
[12] Kim, W., Lee, J., Park, E., & Kim, S. (2006, August). Advanced mechanism for reducing false alarm rate in web page defacement detection. In The 7th International Workshop on Information Security Applications.
[13] Medvet, E., Fillon, C., & Bartoli, A. (2007, August). Detection of web defacements by means of genetic programming. In Third International Symposium on Information Assurance and Security (pp. 227-234). IEEE.
[14] Bartoli, A., Davanzo, G., & Medvet, E. (2010). A framework for large-scale detection of Web site defacements. ACM Transactions on Internet Technology (TOIT), 10(3), 1- 37.
[15] Davanzo, G., Medvet, E., & Bartoli, A. (2011). Anomaly detection techniques for a web defacement monitoring service. Expert Systems with Applications, 38(10), 12521-12530.
[16] Borgolte, K., Kruegel, C., & Vigna, G. (2015). Meerkat: Detecting website defacements through image-based object recognition. In 24th USENIX Security Symposium (USENIX Security 15) (pp. 595-610).
[17] Apruzzese, G., Colajanni, M., Ferretti, L., Guido, A., & Marchetti, M. (2018, May). On the effectiveness of machine and deep learning for cyber security. In 2018 10th international conference on cyber Conflict (CyCon) (pp. 371-390). IEEE.
[18] Abubakar, A., & Pranggono, B. (2017, September). Machine learning based intrusion detection system for software defined networks. In 2017 seventh international conference on emerging security technologies (EST) (pp. 138-143). IEEE.
[19] Calzavara, S., Focardi, R., Squarcina, M., & Tempesta, M. (2017). Surviving the web: A journey into web session security. ACM Computing Surveys (CSUR), 50(1), 1-34.
[20] Sudhodanan, A., Carbone, R., Compagna, L., Dolgin, N., Armando, A., & Morelli, U. (2017, April). Large-scale analysis & detection of authentication cross-site request forgeries. In 2017 IEEE European symposium on security and privacy (EuroS&P) (pp. 350-365). IEEE.
[21] Fernandez, K., & Pagkalos, D. (2017). XSS (Cross-Site Scripting) information and vulnerable websites archive. XSSed. com. Accessed, 14.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY