iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.60646

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 12 Issue IV Apr 2024- Available at www.ijraset.com

Detection of Exploit Websites Using Machine Learning and Data Analysis

Shwetha T.J[1], Deeksha Shukla A[2], Shubham Kumar[3], Simran Das[4]
[1]*Assistant Professor, Department of CSE Impact College of Engineering and Applied Sciences, Bangalore, Affiliated to VTU*
[2, 3, 4]*Students, Department of CSE Impact College of Engineering and Applied Sciences, Bangalore, Affiliated to VTU*

Abstract: ***With the growing integration of internet integration into daily life, the prevalence of security threats such as phishing attacks has become a significant concern. Phishing websites aim to illicitly acquire sensitive user information, posing serious risks to online security. In response, machine learning methods have been investigated for the detection of such fraudulent websites. concentrates on developing a robust phishing website detection system, emphasizing efficiency, accuracy, and cost-effectiveness. Four supervised classification models, namely K-Nearest Neighbour, Kernel Support Vector Machine, Decision Tree, and Random Forest Classifier, are employed and compared. Among these models, the Random Forest classifier demonstrates superior performance, achieving an impressive accuracy score of 96.82% on the selected dataset.***
Keywords: Machine learning techniques, Xgboost, Gradient boosting, Adaboost, SVM, Random Forest, evaluation.

I. INTRODUCTION

Phishing stands out as the foremost prevalent form of social engineering and cyber-attack, where perpetrators target unsuspecting online users, coaxing them into disclosing sensitive information for fraudulent purposes. This nefarious tactic has become a paramount concern for security researchers, given its ease of execution, particularly in the creation of counterfeit websites that closely resemble legitimate ones. While experts may discern these fake websites, many users fall victim to phishing attacks due to their inability to identify them accurately. Primarily, attackers aim to pilfer bank account credentials through such means. The success of phishing attacks is largely attributed to a lack of user awareness. Exploiting vulnerabilities inherent in users, mitigating these attacks proves challenging, underscoring the critical need to enhance phishing detection techniques. Phishing represents a form of widespread fraud, wherein malicious websites masquerade as authentic ones, with the goal of acquiring sensitive information such as passwords, account details, or credit card numbers. Despite efforts to combat phishing through anti-phishing software and detection methods in emails and websites, perpetrators continuously innovate new techniques to circumvent existing defences. Leveraging a blend of social engineering and technology, phishing schemes present themselves as credible communications from reputable entities like financial institutions or e-commerce platforms, enticing users to visit fraudulent websites via provided links. To address this menace, predictive algorithms are being developed to analyze a variety of blacklisted and legitimate URLs, discerning the subtle features that distinguish phishing websites, including those that emerge on short notice (zero-hour phishing websites). Employing ML models and deep neural networks trained on datasets containing both phishing and benign URLs, this project aims to predict and thwart phishing attempts effectively. Web phishing remains a persistent threat, seeking to compromise individuals' private information, emphasizing the ongoing necessity for proactive measures to safeguard against it.

II. LITRETURE SURVEY

The essential centre spins around the thought that authentic websites tend to have various interconnects, authorizing them as dependable. The database stores screenshots and CSS codes of such websites, with CSS serving as a characterizing component of a website's visual components. Aggressors regularly steal bona fide CSS to reproduce a true-blue site's appearance. Subsequently, distinguishing a phishing location includes recognizing a location that mirrors the visual plan or CSS of a veritable location, divulging its pernicious aim in the handle. Eminently, websites connected to at slightest one other location are archived in a whitelist, assuming their authenticity [1]. The method proposed utilizes characteristics derived from the source URL for determining the authenticity of the provided URL. The c4.5 algorithm was employed to formulate the guidelines, subsequently implemented categorize the input URL as either genuine or phishing with improved efficiency. The overall accuracy is limited because of the consideration of the restricted number of URL features [2]. A development of a web scraping tool was designed to gather data from legitimate and phishing websites. An analysis was conducted on the collected data to determine the heuristics rate and the level of engagement towards identifying the legitimacy of a website.

Utilizing a data mining software, the information taken from web scraper was evaluated, leading to the identification of patterns. The model's precise accuracy remains unspecified [3]. The method suggested uses characteristics derived from URL to determine the authentication of the input URL. To establish the guidelines, the c4.5 algorithm was implemented. The generated guidelines are employed to classify the provided URL as legitimate or phishing more efficiently. The lower accuracy and assumption of the dataset regarding legitimate websites are deemed accurate [4]. New Approach utilizing a 2-Phase Detection Model: 1. An ensemble methodology is formulated to authenticate the legitimacy of phishing data and reduce the need for manual labelling. Active learning techniques are employed to achieve this goal.2. The detection model is then trained using the verified data. Issues with the blacklist were identified during monitoring conducted every 12 hours [5]. The three main stages in this process include parsing, heuristic data categorization, and performance assessment within this framework. Each of these steps employs varied and unique approaches for data manipulation to achieve enhanced outcomes. However, incomplete details are provided regarding the methodologies utilized [6]. If tally of occurrences of the root node is: 1. Greater than half of all the nodes, then the probability of genuineness is higher.2. One-quarter of all nodes, the chance of authenticity is average.3. Below one-quarter of the total nodes, then probability of authenticity is low indicating a high risk of phishing. The rates of incorrect negatives and incorrect positives are elevated [7]. Concept of phishing detection method: First, create a dynamic indexing system akin to FVV. Then, utilize adaptive feature selection alongside a self-learning OFSNN model. Integrate multi-modal detection, real-time threat intelligence, explainable AI, and privacy techniques. Augment with data expansion, collaborative learning, and user behaviour analysis for holistic security [8]. The suggested approach employs Fuzzy Rough Set (FRS) theory to identify features, determining decision boundaries through lower and upper approximation regions. By utilizing membership values from these regions, categorization of set members is conducted. However, specific features utilized in the method are not explicitly outlined [9]. The suggested approach involves three stages:1. Extraction and utilization of character succession features from URLs for rapid characterization.2. Utilization of LSTM (Long Short-Term Memory) networks to capture contextual semantics and dependency features of URL character sequences.3. Classification of the extracted features using Soft-max. Yet, this method requires thorough computational resources, rendering it costly [10]. The WCPAD (Web Crawler-based Phishing Attack Detection) comprises three phases:1. Blacklist of DNS.2. Heuristic-based approach.3. Web crawler-based approach, employed for both feature extraction and phishing attack detection. However, the technique is time consuming as each website undergoes all three phases, prolonging the detection process [11]. The approach incorporates a CNN module to extract spatial features from the character-level representations of URLs. These characteristics are then consolidated using a three-layer CNN to generate refined feature representations. However, despite this process, the classifier for phishing URLs experiences a high false positive rate, presenting a significant challenge [12]. A phishing detection system was devised utilizing the XCS classifier, an online adaptive machine learning technique. This method generates numerous rules, termed classifiers. Extracting 38 features from webpage source code and URLs, the model enhances its ability to detect phishing attempts effectively [13].

III. AIM AND OBJECTIVES

The aim and Objectives of this paper are:

1) Employ machine learning methodologies like XgBoost, Gradient Boosting, Adaboost, SVM, and Random Forest for web data mining.
2) Analyze website features like URL structure, domain age, and content to identify patterns indicative of terrorist activity.
3) Rigorously evaluate the model's performance to attain high precision in distinguishing between legitimate and terroristassociated sites.
4) Implement proactive measures such as site blocking based on detection outcomes to thwart the online spread of terrorism.
5) Contribute to enhancing cybersecurity and global security efforts by mitigating the threat posed by terrorist propaganda and activities on the internet.

IV. DESIGN AND IMPLEMENTATION

A. *Proposed System*

By utilising machine learning algorithms, the suggested solution for identifying phishing websites aims to improve upon the flaws in the existing setups. Using supervised learning techniques, which do not necessitate pre-labelled data for training, is one strategy. This method can improve scalability while reducing the costs and effort related to data labelling. Employing multiple machine learning algorithms in tandem is another tactic to increase the system's accuracy and robustness. Furthermore, the system can be enhanced with additional functionalities such as user behaviour tracking and website behaviour analysis, which would improve its ability to detect phishing websites. In conclusion, the suggested solution utilizes machine learning methodologies to improve the accuracy, speed, and effectiveness of phishing website identification.

B. Modules A. User

1) Home Page: The phishing website prediction web application's home page is accessible to users.
2) About Page: gives details on the feature that detects phishing attempts.
3) View Dataset: To view the dataset, users should go to this website.
4) Input Values: In order to receive results, users must provide values for certain fields.
5) Results Page: The model's generated results are visible to users.
6) Score Page: Provides a percentage score for users to examine. B. Framework:
7) Data Availability Check: The system loads data into CSV files after confirming that it's available.
8) Preparing the Data: pre-processing the information within accordance with the models in order to improve data insights and model accuracy.
9) Splitting Training Data: Before using specific methods for training, data is divided into training and testing sets.
10) Model Construction: Aids in creating a model for more accurate phishing activity prediction.
11) Score Generation: Gives consumers accessibility to the percentage score that has been generated.
12) Result Generation: Phishing activity is predicted by machine learning algorithms that possess been trained.
13) Produce Results: The system computes the personality prediction following the training of the machine learning model algorithm.

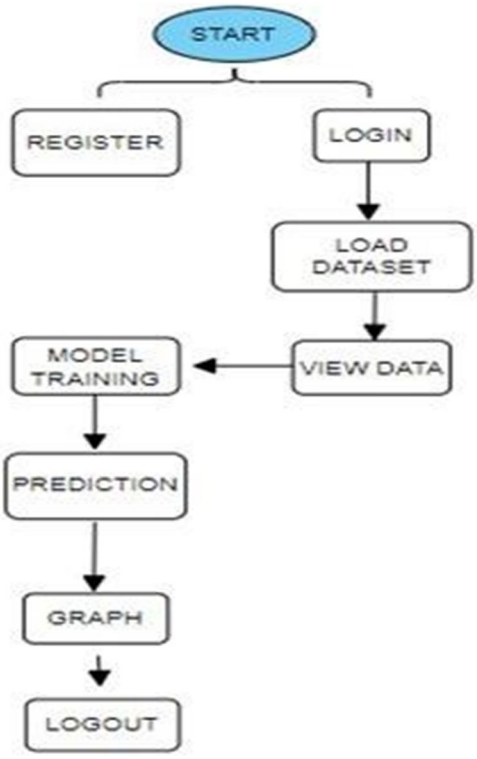


Fig 1: Proposed System Architecture

V. METHODOLOGY

A. ADABOOST Classifier

AdaBoost stands as a prominent method within the realm of machine learning, where it amalgamates weak learners to bolster decision-making prowess. This technique operates through a sequential training process, whereby each subsequent learner prioritizes rectifying the errors of its predecessors. This systematic method not only elevates accuracy but also mitigates errors effectively. Particularly adept at refining decision trees, AdaBoost, initially labelled AdaBoost.M1 and now often referred to commonly as discrete AdaBoost, excels in binary classification endeavours such as identifying spam. While its primary application lies in classification rather than regression tasks, AdaBoost exhibits a knack for enhancing the efficacy of various machine learning models, particularly when dealing with weak learners.

B. XGBOOST

XGBoost, abbreviated from Extreme Gradient Boosting, represents a highly optimized library for distributed gradient boosting well-regarded for its efficiency, versatility, and adaptability across platforms. Operating within the framework of Gradient Boosting, it employs machine learning algorithms to provide parallel tree boosting algorithm capabilities, swiftly addressing diverse data science challenges with precision. Boosting, an ensemble learning method, sequentially assembles a robust classifier from weaker ones, effectively managing the trade-off between bias and variance. Unlike bagging algorithms, boosting effectively regulates both bias and variance, resulting in notable efficacy. Beyond XGBoost, notable boosting techniques encompass AdaBoost, Gradient Boosting, CatBoost, and Light GBM. XGBoost, a continuation of gradient boosted decision trees, has emerged as a leading solution renowned for its scalability, dominating realms such as applied machine learning and Kaggle competitions, particularly excelling with structured data because of exceptional speed and performance.

C. Random Forest Classifier

A Random Forest, categorized as an ensemble learning method, integrates numerous decision trees to effectively tackle regression and categorization tasks. It constructs the forest using bagging, which enhances accuracy through bootstrap aggregating. By averaging the predictions of individual trees, the algorithm makes predictions, with precision scaling alongside the quantity of trees. Random Forest effectively addresses the limitations often observed in decision trees, notably overfitting, and demonstrates competence in managing missing data without necessitating extensive hyper-parameter tuning. At each node split, every tree randomly selects a segment of features, with these nodes representing attributes utilized for prediction. Decision trees, consisting of decision, leaf, and root nodes, serve as the fundamental components of a random forest, partitioning data until reaching homogeneous leaf nodes.

D. Gradient Boosting Classifier

Gradient Boosting, acknowledged as a robust machine learning method, effectively alleviates bias error within models. Unlike AdaBoost, it utilizes a predetermined base estimator, typically Decision Stumps. While users retain the option to adjust the estimator parameter, the default value typically stands at 100. This algorithm showcases adaptability, proficiently predicting both continuous (as a Regressor) and categorical (as a Classifier) target variables. In regression scenarios, it optimizes the Mean Square Error (MSE) cost function, whereas in classification tasks, it employs the Log loss function. Noteworthy is Gradient Boosting's prowess in mitigating bias error and demonstrating proficiency across various predictive tasks with remarkable precision.

E. Support Vector Machine (SVM)

This algorithm aims to identify a hyperplane in an N-dimensional space to distinctly classify data points, maximizing the margin between different classes. Support vectors, crucial points influencing the hyperplane's position and orientation, aid in this process. Removing them alters the hyperplane's position, underscoring their significance in building the SVM model.

F. System Design

1) Input Design

During the design phase, input plays a crucial role in information systems. Input essentially consists of the initial data that undergoes processing to yield output. When developers focus on input design, they must consider various input devices like PCs, MICRs, OMRs, and others. As a result, the quality of input directly impacts the quality of output in a system. A well-constructed input design exhibits specific characteristics: Effective fulfilment of a designated purpose, such as information storage, recording, and retrieval. Ensuring accurate and complete data entry. Ease of use and simplicity in filling out forms. Prioritizing user attention, consistency, and straight forwardness. These objectives rely on fundamental design principles concerning the required system inputs and understanding how end users interact with different elements on forms and screens.

a) *Objectives For Input Design:* The Goals of Input Design include: Designing data entry and input procedures. Decreasing input volume. Creating source documents for data capture or developing alternative data capture methods. Designing input data records, data entry screens, user interface screens, etc. Implementing validation checks and establishing efficient input controls.

2) Output Design

The creation of yields stands out as a crucial obligation in any framework. Engineers lock in in yield plan by recognizing the specified yield sorts and mulling over the fundamental yield controls and model report courses of action.

a) *Objectives Of Output Design:* The Goals of Input Design: Developing output design that fulfils the intended purpose and avoids generating unnecessary output. Meeting the end user's requirements through appropriate output design. Ensuring the right quantity of output is delivered. Formatting the output correctly and directing it to the intended recipient. Timely availability of output to facilitate sound decision-making.

VI. CONCLUSION

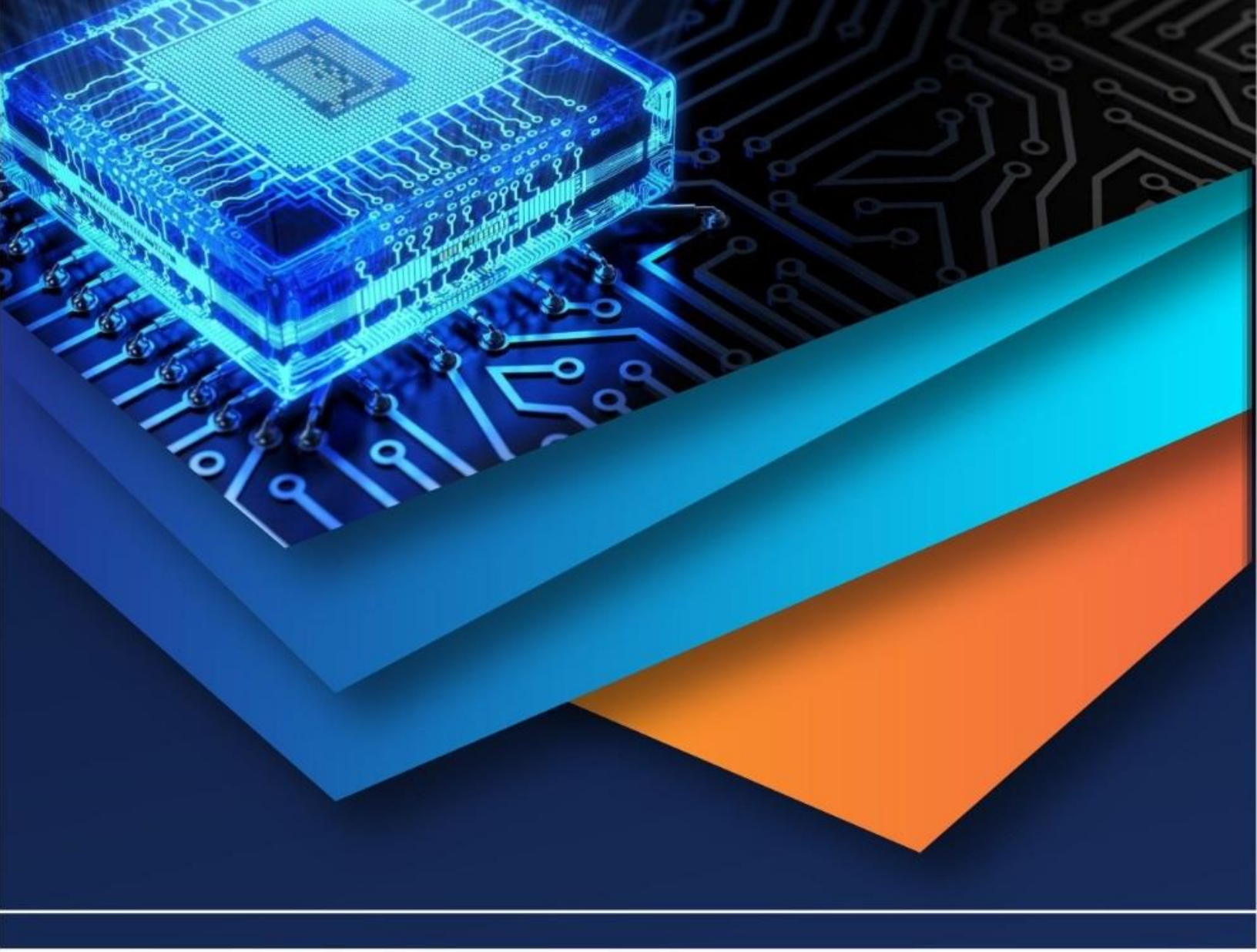
In conclusion, web data mining combined with machine learning techniques presents a viable strategy to counter the spread of terrorism online. Researchers can use algorithms like XgBoost, Gradient Boosting, Adaboost, SVM, and Random Forest to analyse various aspects of websites and differentiate between websites that are linked to terrorist activity and those that are not. By streamlining the detection process, these methods increase efficiency and scalability while lowering the requirement for manual intervention. Their remarkable accuracy rates have been proven by extensive examination, outperforming traditional rulebased techniques. Utilizing machine learning-driven detection systems has the potential to significantly improve cybersecurity defences against online terrorist activity and propaganda. It is imperative to pursue ongoing research and progress in this field to remain ahead of the ever-evolving strategies utilised by terrorist groups.

VII. ACKNOWLEDGMENT

We are thrilled to have successfully completed our task and deeply appreciate the support and guidance we've received throughout. Our connection with the Impact College of Engineering and Applied Sciences community has been an unwavering source of encouragement. We extend our heartfelt thanks to Mrs. Shwetha T.J, Assistant Professor in the Department of Computer Science and Engineering, for her diligent review and correction of our documents. Special recognition is reserved for Dr. Dhananjaya V, Professor and Head of the Department of Computer Science and Engineering, for his invaluable guidance and support. Our gratitude also extends to the Management and Principal, Dr. Jalumedi Babu, for their steadfast support. Acknowledgment is also owed to the faculty members and support staff of the Department of Computer Science and Engineering for their valuable assistance in completing Phase One of our Major Project. Lastly, we wish to express our sincere appreciation to our parents and friends for their continuous support during the initial phase of our major project.

REFERENCES

[1] S. Haruta, H. Asahina, and I. Sasase, Visual Similarity-based Phishing Detection Scheme, 2017.
[2] Lisa Machado and Jayant Gadge, Phishing Sites Detection using C4.5 Decision Tree Algorithm, 2017.
[3] A. J. Park, R. N. Quadari, and H. H. Tsang, Phishing Website Detection Framework through Web Scraping and Data Mining, 2017.
[4] H. Shirazi, K. Haefner, and I. Ray, Fresh-Phish, 2017.
[5] J. Li and S. Wang, Phishbox,2017.
[6] S. Parekh, D. Parikh, S. Kotak, and S. Sankhe, Detection of Phishing Websites through URL Analysis, 2018.
[7] C. E. Shyni, A. D. Sundar, and G. S. E. Ebby, OFs-NN, 2019.
[8] E. Zhu, Y. Chen, C. Ye, X. Li, and F. Liu Ofs-nn: An Effective Phishing Website Detection Model, 2019.
[9] Mahdieh Zabihimayvan and Derek Doran, Fuzzy Rough Set Feature Selection for Enhancing Phishing Attack Detection, 2019.
[10] P. Yang, G. Zhao, and P. Zeng, Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning, 2019.
[11] T. Nathezhtha, D. Sangeetha, and V. Vaidehi, Wc-pad: Web Crawling-based Phishing Attack Detection, 2019.
[12] Y. Huang, Q. Yang, J. Qin, and W. Wen, Phishing URL Detection via CNN and Attention-based Hierarchical Run, 2019.
[13] M. M. Yadollahi, F. Shoeleh, E. Serkani, A. Madani, and H. Gharaee, Adaptive Machine Learning-Based Approach for Phishing Detection Using Hybrid Features, 2019.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY