IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DETECTION OF MALICIOUS UNIFORM RESOURCE LOCATOR USING MACHINE LEARNING

Saptaparni Chatterjee, Joydeep Mukherjee,

Student, Assistance Professor,
School of Education Teconology ,
Jadavpur University, Kolkata, India

Abstract: Now-a-days people are unaware of the fact that malicious webpages have such kind of scripts that they don't have to intentionally download a malicious attachment in order to compromise computer security,i.e.;it gets automatically downloaded in system. To saveimportant data and software from these types of threats we must use some updated algorithm to detect malicious webpage .In the present research a machine learning based approached is proposed to identify malicious webpages from a heap of URL data. Random forest, logistic regression, supportvector machine, K-Nearest Neighbors Algorithm is used. The data heap used in this research is collectedfrom well-known sources such as GitHub, Kaggle etc. This novel work is further valuated against traditional machine learning model and our result highlights positive state in respect to the proposed approach. Hyper -parameter tuning, Quantile transformer make algorithm unique and classification, confusion matrix, ROC curve give theresults of evaluation.

Index Terms - Machine learning, URL, Logistic regression, Random Forest, Support Vector Machine, K-Nearest Neighbors Algorithm, Confusion matrix, ROC curve, Classification Report, Hyper-parameter tuning, Quantile transformer.

I. INTRODUCTION

The internet plays a crucial role in our daily lives, enabling us to connect, communicate, and access information easily. However, a downside to thisconvenience is the proliferation of rogue Uniform Resource Locators (URLs) that pose serious threats to users' security and privacy. Malicious URLs can leadto various cybercrimes such as phishing attacks,identity theft, and malware dissemination. To addressthese dangers, it is imperative to develop effective methods to identify and counteract them. Machine learning techniques, such as Support Vector Machines, K-Nearest Neighbors Algorithm, logistic regression, and random forest, have shown promising results in accurately identifying fraudulent URLs. This thesis aims to contribute to cybersecurity by investigating the use of random forest in identifyingdangerous URLs.

The problem statement arises from the constant evolution of cyber threats, making it challenging todetect harmful URLs using traditional heuristic and signature-based techniques. A more sophisticated and adaptable approach is needed toeffectively identify and categorize malicious URLs. The objective of this study is to construct areliable and efficient system for detecting malicious URLs using machine learning, particularly the random forest algorithm. The study includes tasks such as testing the performance of machine learning algorithms in spotting malwareURLs, creating a dataset of both malicious and benign URLs for training and testing, extracting relevant information from URLs to differentiate between harmful and benign ones,

implementing a random forest model for detection, and evaluating its accuracy and performance metrics compared to other existing machine learning techniques.

The motivation behind this research stems from the growing concern among individuals, organizations, and governments about the increasing frequency and sophistication of cyberattacks. Malicious URLs have become a prevalent vector for launching these attacks, posing significant risks to users' online activities and sensitive data. By leveraging machine learning techniques, we can develop sophisticated systems capable of analyzing a vast amount of URLs and accurately identifying potential risks. This thesis aims to advance cybersecurity practices and mitigate the threats posed by online attacks by contributing to the development of reliable and efficient methods for identifying malicious URLs.

II. RELATED WORK:

Alshammari et al. conducted a comparative analysis of machine learning algorithms for detecting malicious Uniform Resource Locators. They evaluated the performance of various algorithms, including Random Forest, Naive Bayes, K-Nearest Neighbors, Support Vector Machine, and Logistic Regression. The study found that Random Forest outperformed other algorithms in terms of accuracy, precision, recall, and F1-score [2]. Grosse et al. explored the vulnerability of deep neural networks (DNNs) to adversarial examples in the context of malware classification. They demonstrated that by manipulating certain features of malicious Uniform Resource Locators, it is possible to evade detection by DNN-based classifiers. The study highlights the need for robust defenses against adversarial attacks in the domain of malicious Uniform Resource Locator detection [3].Xiang et al. proposed a deep learning framework for malicious Uniform Resource Locator detection based on an attention mechanism. The model utilizes a long short-term memory (LSTM) network with an attention mechanism to capture important features from Uniform Resource Locators. The study demonstrated that the proposed framework achieved competitive performance compared to traditional machine learning algorithms[4].Ahmadi and Zolanvari investigated the use of deep transfer learning for malicious Uniform Resource Locator detection. They proposed a transfer learning framework based on pre-trained convolutional neural networks (CNNs) to leverage knowledge from related domains. The study showed that the proposed approach improved the performance of malicious Uniform Resource Locator detection compared to traditional machine learning algorithms [5].Cao et al. presented DeepTransfer, a deep neural network for cross- domain malicious Uniform Resource Locator detection. The model utilizes a CNN-based architecture combined with transfer learning techniques to detect malicious Uniform Resource Locators across different domains. The study demonstrated the effectiveness of Deep Transfer in achieving high detection accuracy and outperforming traditional machine learning algorithms[6].Arachchilage and Love conducted a survey on feature selection and extraction methods for malicious Uniform Resource Locator detection with a focus on explainable artificial intelligence (XAI). The study provides an overview of different techniques and highlights the importance of interpretable models to understand the decision-making process of Uniform Resource Locator classifiers [7].Zhu et al. proposed a novel malicious Uniform Resource Locator detection system based on an improved CNN. They introduced a dual- channel CNN architecture that incorporates both Uniform Resource Locator text and structure information. The study demonstrated that the proposed system achieved high detection accuracy and outperformed traditional machine learning algorithms [8].Chang et al. presented a hybrid model for malicious Uniform Resource Locator detection based on gradient boosting decision trees (GBDT) and multilayer perceptron (MLP). The model combines the strengths of both algorithms to improve detection accuracy. The study showed that the hybrid model outperformed individual algorithms and achieved competitive performance in malicious Uniform Resource Locator detection[9].

III. WORKING METHODOLOGY:

Machine analysis aims to thoroughly understand the workings of a machine and identify areas for improvement, while system analysis focuses on clearly characterizing the technical components of a system. Numerous machine learning methods, such as the K-Nearest Neighbors Algorithm, logistic regression, Support Vector Machine, and random forest, are included in the suggested design. The system incorporates these algorithms. The preprocessing of the data is the first step, and model training is the next. The performance of the models is improved by using methods like hyper-parameter tuning and the quantile transformer. The Random Forest model stands out as the most accurate algorithm among them all, offering trustworthy forecasts. Through a function call, users can provide data for predictions, and the system will use the trained model to produce accurate results.

The software implementation focuses on classifying Uniform Resource Locators (URLs) as either good or bad based on various features. The implementation includes modules for data preparation, feature engineering, data splitting,model selection, and training using machine learning algorithms such as Random Forest, Logistic Regression, Support Vector Machine,and K-Nearest Neighbors. The models are evaluated using metrics like accuracy, true positive rate, false positive rate, precision, recall, F-measure, and area under the ROC curve. Visualization techniques are used to analyze the model's performance. The Random Forest algorithm is described, highlighting its ability to aggregate predictions from multiple decision trees, reduce errors, handle complex datasets, and generalize well to unseen data. Data visualization and preprocessing techniques are employed to ensure data quality, completeness, and handle missing values appropriately, considering their impact on bias and analysis results. Understanding the patterns and reasons behind missing data is essential for minimizing bias and making informed decisions during data analysis. The performance of the models is improved by using methods like hyper-parameter tuning and the quantile transformer.
'url', 'label', and 'type' are the three columns in the dataset, which has 691 items total.
The Uniform Resource Locators are present in the column 'url' as objects.

The labels for the Uniform Resource Locators are included in the 'label' column and include terms like 'good' or 'bad'.
The integer column with the values 0 and 1 in the 'type' column.An unbalanced dataset is indicated by the 'type' column, which contains counts of 184 for value 0 and 507 for value 1. There are no blank cells in any of the columns of the dataset.The dataset has undergone further pre-processing and feature engineering operations after the addition of the extra features.

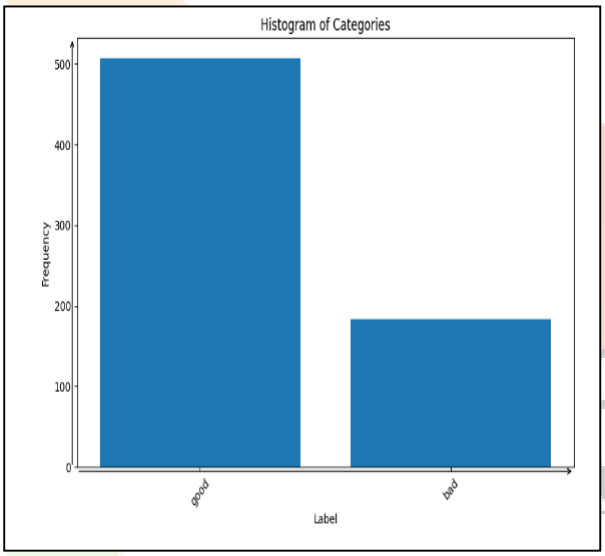

fig1 histogram of catagories of dataset

This fig1 histogram of categories showing the amount of bad url and good url in my dataset.

Data processing tasks, such as checking null values and missing values, ensure data quality, completeness, and reliability, improving the accuracy and validity of data analysis and machine learning models. Feature engineering involves selecting relevant features using chi-square tests, which examine the relationship between URL categories and different URL features.
The system implementation includes details of each feature, such as abnormal URL, count of dots, count of "www," shortened URL, character counts, URL length, hostname length, and more. Various classification models like Random Forest, Support Vector Machines, Logistic Regression, and K- Nearest Neighbors are used, along with hyperparameter optimization and evaluation metrics like precision, recall, F1-score, and confusion matrix. The software implementation combines these techniques to create a comprehensive solution for URL classification.

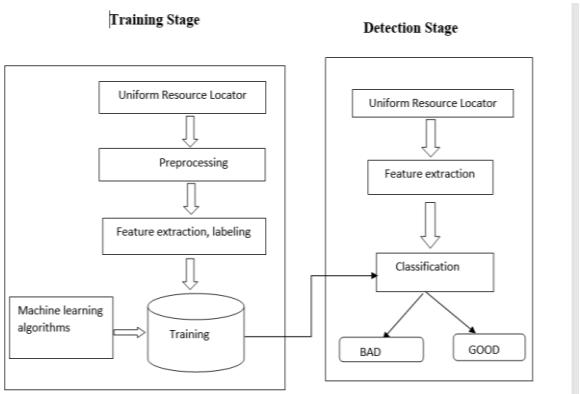

fig 2 architecture

This fig2 is a architecture of the full process how this is executed.
The quantile transformer and hyperparameter tuning are two techniques commonly used in machine learning to improve model performance.

1. Quantile Transformer:

The quantile transformer is a data preprocessing technique that transforms the features of a dataset to follow a uniform or Gaussian distribution. It maps the original data to a new space where the quantiles of the transformed data are uniformly distributed or follow a Gaussian distribution. This transformation helps to mitigate the impact of outliers and non- normality in the data, which can affect the performance of certain machine learning algorithms.

Mathematically, the quantile transformer computes the quantile function of each feature, which maps the original feature values to quantiles. It then applies a non-linear transformation to the quantiles to obtain the transformed values. The transformation can be performed using different approaches, such as the Gaussian or uniform transformation. The resulting transformed data is then used as input for the machine learning model.

2. Hyperparameter Tuning:

Hyperparameters are parameters that are not learned from the data but are set prior to training the model. Examples of hyperparameters include the learning rate, regularization strength, number of hidden layers in a neural network, etc. Hyperparameter tuning refers to the process of selecting the optimal combination of hyperparameters to maximize the model's performance.

The mathematical theory behind hyperparameter tuning involves searching for the best set of hyperparameters that minimizes a chosen evaluation metric, such as accuracy or mean squared error. This is typically done using optimization techniques like grid search, random search, or more advanced methods like Bayesian optimization. These techniques explore the hyperparameter space, evaluating the model's performance with different hyperparameter configurations, and selecting the combination that yields the best results.

In summary, the quantile transformer is a mathematical transformation that maps the original feature values to quantiles, helping to normalize the data. Hyperparameter tuning, on the other hand, involves searching for the optimal combination of hyperparameters that maximizes the model's performance. Both techniques aim to improve the accuracy and effectiveness of machine learning models.

The models mentioned in the description include Random Forest, Support Vector Machines (SVM), Logistic Regression, and K-Nearest Neighbors (KNN) Algorithm. Random Forest is an ensemble learning method that combines multiple decision trees, SVM finds an optimal hyperplane to separate classes, Logistic Regression models the probability of binary outcomes, and KNN assigns labels based on the majority vote of nearest neighbors.

To evaluate the performance of these models, several metrics are commonly used. The classification_report function provides metrics such as precision, recall, F1-score, and support for each class. Precision measures the

accuracy of positive predictions, recall measures the ability tocorrectly identify positive instances, and the F1-score balances precision and recall. These metricsoffer insights into the model's performance for different classes. Additionally, theconfusion_matrix function generates a matrix showing true positives, false positives, truenegatives, and false negatives, providing a detailed breakdown of the model's performance.

Accuracy is another important metric calculated by dividing the sum of true positives and true negatives by the total number of samples. It provides an overall measure of the model's performance. The Receiver Operating Characteristic (ROC) curve graphically representsthe trade-off between true positive rate and false positive rate at various thresholds model's discriminative The area under the ROC curve (AUC) indicates the ability, with a value of 1 denoting flawless classification.

These metrics and evaluation techniques play a crucial role in assessing and comparing the performance of classification models, aiding in model selection, and identifying areas for improvement.
Mathematically, precision can be expressed as:Precision = TP / (TP + FP)
Mathematically, recall can be expressed as:Recall = TP / (TP + FN)

Mathematically, the F1-score can beCalculated as:
F1-score = 2 * (Precision * Recall) / (Precision +Recall).

IV. RESULTS AND DISCUSSION

TheproposedworkistestedonWindows11operating system, with 8 GB RAM, and a Intel i3processor.The programming language used to performexperimentation is Python.

Classifier: Random Forest

	precision	recall	f1-score	support
bad	1.00	0.97	0.99	37
good	0.99	1.00	1.00	102
accuracy			0.99	139
macro avg	1.00	0.99	0.99	139
weighted avg	0.99	0.99	0.99	139

---

fig 3 random forest classification

fig3 represent the classification report ofRandom Forest.

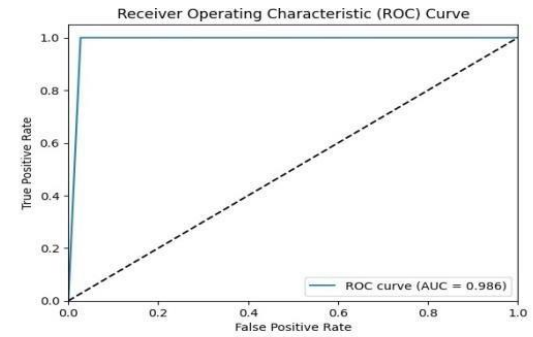

fig4 represent the roc curve ofrandom forest.

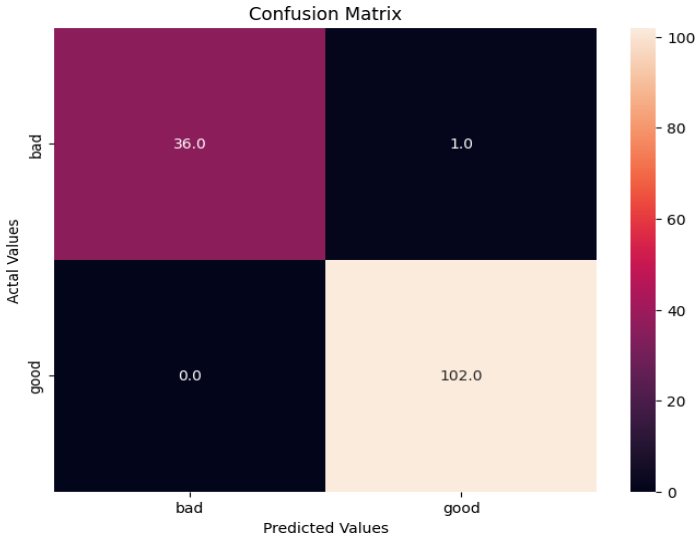

fig 5 confusion matrix

this fig 5 represent the confusionmatrix of random forest algorithm.

i table of comparison

ALGORITHM	Existing approach	AFTER HYPERPARAMETER TUNNING,QUANTILE TRANSFORMER,CHI-SQUARE TEST ACCURACY (proposed approach)
LOGISTIC REGRESSION	0.90	0.96
K-Nearest Neighbors Algorithm	0.90	0.971
Support Vector Machine	0.94	0.986
RANDOM FOREST	-	0.993

This table represents the comparison between existing approaches accuracy and this new approach accuracy, Random Forest has the best accuracy 0.993%

V. **CONCLUSION:** The system implementation focuses ondeveloping a URL classification model using machine learning techniques. It involves data preparation, feature engineering, model selection and training, prediction, and evaluation. Algorithms such as Random Forest, Logistic Regression, Support Vector Machine, Neural Network, and K-Nearest Neighbors are utilized for classification.

Features include characteristics like dot count, www count, presence of shortening service, URL length, digit and letter count, and subdomain count. Chi-square tests are used for feature selection and validation, considering the correlation between URL classification and features. Performance evaluation includes accuracy, true positive rate, false positive rate, precision, recall, F1-score, and area under the ROC curve. Visualizations like ROC curves and accuracy plots provide insights. Hyperparameter tuning, optimization, regularization, and quantization are emphasized for achieving optimal accuracy. Classification reports and confusion matrices analyze the model's performance in detail.Future scope of this work Dataset Enhancement: Larger and diverse datasets, broader URL coverage, and improved detection accuracy.Advanced Feature Engineering: Explore additional features, feature selection techniques, and identification of relevant features.Ensemble Methods: Implement stacking or boosting for improved performance and bias mitigation.Real-Time Implementation: Adapt the model for real-time URL classification and developscalable systems.Deployment and Integration: Integratethe model into existing security systems and web browsers, providing instant classification results.

VI. ACKNOWLEDGMENT

I feel extremely glad in publishing this research .It is a great pleasure for me to express my respect and a deep sense of gratitude to my family continuous support during the entire course of research work. Their advice and support were highly inspirational and motivating.

VII.References

[1]RupaChiramdasu,GautamSrivastava,SwetaBhattacharya, Praveen Kumar Reddy,Thippasandra Reddy Gadekallu
,"Malicious URL Detection using Logistic Regression", in 2021 IEEE Conference on Omni-Layer Intelligent Systems (COINS) .IEEE,2021,pp. 1-6.
[2]R. Alshammari, G. Alshammari, and K. Elleithy, "A Comparative Analysis of Machine Learning Algorithms for Detecting Malicious URLs," Journal of Information Security and Applications, vol. 52, p. 102504, 2020.
[3] K. Grosse, J. Saxe, and R. Yerneni, "Adversarial Examples Against Deep Neural Networks in Malware Classification," in 28th USENIX Security Symposium (USENIX Security 19), 2019, pp. 577-594.
[4]T. Xiang, X. Zhang, and J. Dong, "A Deep Learning Framework for Malicious URL Detection Based on Attention Mechanism," Complexity, vol. 2020, pp. 1-10, 2020.
[5]M. Ahmadi and A. Zolanvari, "Deep Transfer Learning for Malicious URL Detection," Soft Computing, vol. 25, no. 12, pp. 8099-8110, 2021.
[6]Y. Cao, X. Liu, and D. Zhang, "DeepTransfer: A Deep Neural Network for Cross-Domain Malicious URL Detection," Journal of Computers, vol. 15, no. 6, pp. 1018- 1028, 2020.
[7]N. A. G. Arachchilage and S. Love, "Towards Explainable Artificial Intelligence (XAI): A Survey on Feature Selection and Feature Extraction Methods for Malicious URL Detection," Computers & Security, vol. 87, p. 101607, 2019.
[8]L. Zhu, C. Xing, and W. Wang, "A Novel Malicious URL Detection System Based on Improved Convolutional Neural Network," Journal of Intelligent & Fuzzy Systems, vol. 37, no. 2, pp. 2053-2063, 2019.
[9]Y. Chang, Y. Lin, and M. Lin, "A Hybrid Model for Malicious URL Detection Based on GBDT and MLP," Future Generation Computer Systems, vol. 110, pp. 764-772, 2020.
[10]D. R. Costa, G. A. França, and J. M. dos Santos,"A Comparative Study of Machine Learning Algorithms for Malicious URL Detection," in Proceedings of the 8th Brazilian Conference on Intelligent Systems (BRACIS), 2019, pp. 337-342. [11]X. Pan, W. Song, and J. Tian, "Malicious URL Detection Based on Hierarchical Attention Network and Convolutional Neural Network," Computers, Materials & Continua, vol. 65, no. 2, pp. 1497-1513,
2020.
[12]S. Kumar and R. Rathee, "URL classification using machine learning techniques: A survey," Artificial Intelligence Review, vol. 47, no. 1, pp. 1-
38, 2017.
[13]H. Keshav, M. Patel, and N. Jain, "Machine learning techniques for detecting malicious URLs: A survey," Future Generation Computer Systems, vol. 82, pp. 481-497, 2018.
[14]S. Maity and P. K. Jana, "Analysis of machine learning techniques for URL classification," in International Conference on Computational Intelligence in Data Science, 2018, pp. 325-337. [15]G. H. Poh and B. K. Tan, "A comparative study on malicious URL detection using machine learning," in 2019 3rd International Conference on Intelligent Sustainable Systems (ICISS), 2019, pp. 524-529. [16]Natarajan, M.,Karthikeyan,S.,&Priyadarshini, S.(2021). Malicious URL Detection using Deep Learning Techniques. In International Conference on Information Science, Computing, and Communication (pp. 383-391). Springer, Singapore. [17]Chaudhary, S., & Shukla, A. (2020). Malicious URL Detection using Machine Learning Techniques. International Journal of Computer Sciences and Engineering, 8(6), 192-197.
[18] Hu, W., & Hu, M. (2020). Malicious URL Detection using Deep Learning and Feature Engineering. Journal of Supercomputing, 76(6), 4396- 4415.
[19] Sivakumar, R., Dhamodharan, M., &Yogesh, S. (2020). Analysis of Machine Learning Algorithms for Malicious URL Detection. International Journal of Innovative Technology and Exploring Engineering, 9(1), 1906- 1910.
[20] Jain, S., & Chand, R. (2020). Comparative Analysis of Machine Learning Algorithms for Malicious URL Detection. International Journal of Computer Applications, 174(2), 11-16.
[21] Wagle, M., Bajracharya, P., &Gurung, P. (2020).Comparative Analysis of Machine Learning Algorithms for Malicious URL Detection. In 2020 International Conference on Sustainable

Technologies for Industry 4.0 (pp. 1-5). IEEE.
[22] Nair, A. S., &Murugan, S. (2020). Comparative Analysisof Machine Learning Techniques for Malicious URLDetection. In 2020 International Conference on Communication and Electronics Systems (ICCES) (pp. 85- 89). IEEE.
[23] Padmakumari, P., &Sabu, N. (2020). Malicious URL Detection using Machine Learning Algorithms. In 2020 IEEEInternational Conference on System, Computation, Automation and Networking (ICSCAN) (pp. 321- 325). IEEE.
[24] Zhang, C., &Xue, R. (2019). Comparative Analysis of Machine Learning Algorithms for Malicious URL Detection.In 2019 12th International Conference on Human System Interaction (HSI) (pp. 654-657). IEEE.
[25] Sengar, H. S., Sharma, A., & Sharma, D. K. (2019). Comparative Study of Machine Learning Algorithms forMalicious URL Detection. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 657-660). IEEE.
[26] Pathak, A., & Jain, P. (2019). Comparative Study of Machine Learning Algorithms for Malicious URL Detection. In 2019 5th InternationalConference on Advanced Computing & Communication Systems (ICACCS) (pp. 1220-1224). IEEE.
[27] Yadav, A., Yadav, N., & Sharma, S. K. (2019). Comparative Analysis of Machine Learning Algorithms for Malicious URL Detection. In 2019 6th International Conference on Advanced Computing and Communication Systems (ICACCS) (pp. 1073-1077). IEEE.
[28] Gupta, A., Kumar, S., & Jain, N. (2019). A Comparative Analysis of Machine Learning Techniques for Malicious URL Detection. In 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-6). IEEE.
[29] Kumar, R., Kalia, A., & Singh, R. (2019). Comparative Study of Machine Learning Algorithms for Malicious URL Detection. In 2019 International Conference on Communication and Electronics Systems (ICCES) (pp. 247-251). IEEE.
[30] Solanki, K., Jethva, N., & Patel, R. (2019). Comparative Analysis of Machine Learning Algorithms for MaliciousURL Detection. In 2019 International Conference onInnovative Trends in Computer Engineering (ICITCE) (pp. 1-5). IEEE.
[31] Kaur, J., & Kaur, A. (2019). Comparative Analysis of Machine Learning Algorithms for Malicious URL Detection.In 2019 International Conference on Automation, Computational and Technology Management (ICACTM) (pp. 272-277). IEEE.
[32] Khan, N., Shah, H., &Naik, K. (2019). Comparative Analysis of Machine Learning Techniques for MaliciousURL Detection. In 2019 3rd International Conference on Advances in Electronics, Computers and Communications (ICAECC) (pp. 1-6). IEEE.
[33] Priyadharshini, R., &Bhuvaneswari, V. (2019). Comparative Analysis of Machine Learning Algorithms for Malicious URL Detection. In 2019 International Conferenceon Smart Technologies for Smart Nation (SmartTechCon) (pp. 244-249). IEEE.
[34] Sharma, N., & Jain, N. (2019). Comparative Analysis ofMachine Learning Techniques for Malicious URL Detection.In 2019 International
Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 546- 550). IEEE.
Verma, M., Jaiswal, N., &Tanwani, A. (2019). Comparative Analysis of Machine Learning Techniques for Malicious URL Detection. In 2019 International Conference on Communication, Computing and Internet of Things (IC3IoT) (pp. 1-4).IEEE.
[35] Sharma, S., & Bharti, S. (2019). Comparative Analysis of Machine Learning Techniques for Malicious URL Detection. In 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-6). IEEE.
[36] Gupta, R., Srivastava, P., & Asthana, S. (2019). Comparative Analysis of Machine Learning Algorithms for Malicious URL Detection. In 2019 International Conference on Automation, Computational and Technology Management (ICACTM) (pp. 261-266). IEEE.
[37] Dwivedi, A., & Jain, S. (2019). Comparative Analysis of Machine Learning Techniques for Malicious URL Detection. In 2019 2nd International Conference on Inventive Research inComputing Applications (pp. 1179-1183). IEEE. [39]Yadav, R., &Rathore, S. (2019). Comparative Analysis of Machine Learning Techniques for Malicious URL Detection. In 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 186-190). IEEE. [40]Kumar, R., Kalia, A., & Singh, R. (2019).Comparative Study of Machine Learning Algorithmsfor Malicious URL Detection. In 2019 InternationalConference on Communication and ElectronicsSystems (ICCES) (pp. 247-251). IEEE.
[41] Waghmare, P., & Agarwal, P. (2019). Comparative Analysis of Machine Learning Algorithms for Malicious URL Detection. In 2019 6th International Conference on Computing for Sustainable Global

Development (INDIACom) (pp. 1283-1287). IEEE.

[42] Saxena, N., & Yadav, N. (2019). Comparative Analysis of Machine Learning Techniques for Malicious URL Detection. In 2019 International Conference on Intelligent Sustainable Systems (ICISS) (pp. 549-554). IEEE.

[43] Kandari, M.S., &Sharma, D. (2019). Comparative Analysis of Machine Learning Algorithms for Malicious URL Detection. In 2019 5th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU) (pp. 1-6). IEEE. [44]Jothi, P. R., &Aramudhan, M. (2018). Comparative Study of Machine Learning Techniques for Malicious URL Detection. In 2018 International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT) (pp. 1-6). IEEE.

[45]Nagpal, A., & Kaur, M. (2018). Comparative Analysis of Machine Learning Techniques for Malicious URL Detection. In 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT) (pp. 1-6). IEEE. [46]Pawar, R., &Kolhe, S. (2018). Comparative Analysis of Machine Learning Techniques for Malicious URL Detection. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 553-557). IEEE.

[47] Chakraborty, P., &Chakrabarti, A. (2018). Comparative Analysis of Machine Learning Techniques for Malicious URL Detection. In 2018 International Conference on Computing, Power and Communication Technologies (GUCON) (pp. 165-169). IEEE.

[48] Jaiswal, A., &Patidar, V. (2017). Comparative Analysis of Machine Learning Algorithms for Malicious URL Detection. In 2017 IEEE International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT) (pp. 1-6). IEEE. [49]Sharma, N., & Jain, N. (2017). Comparative Study of Machine Learning Algorithms for Malicious URL Detection. In 2017 2nd International Conference for Convergence in Technology (I2CT) (pp. 85-90). IEEE.

[50]Sahdev, G., & Chhabra, A. (2017). Comparative Analysis of Machine Learning Algorithms for Malicious URL Detection. In 2017 International Conference on Intelligent Sustainable Systems (ICISS) (pp. 620-625). IEEE.